

TD

Diagramme de classes, d'objets et de séquence

Diagramme de classes métier

Design pattern

Sommaire

Sommaire	1
1 - Diagramme de classe	2
1 : La voiture (d'après L. Audibert – Ellipses)	2
2 : Système de fichiers (d'après L. Audibert – Ellipses)	2
3 : Le robot	3
4 : Jeu de rôles et tournevis !	4
5 : L'application compteur	5
6 : Les figures	6
7 : Equation du second degré	6
8 : Tetris	7
2 - Diagramme de classe métier	8
1 : La cinémathèque	8
2 : L'association	8
3 : L'hôtel	9
3 – MVC	10
4 – Interfaces et design pattern	11
1 : Le téléviseur	11
2 : Les canards (DPTLP)	12
3 : Dessiner et imprimer	12

1 - Diagramme de classe

1 : La voiture (d'après L. Audibert – Ellipses)

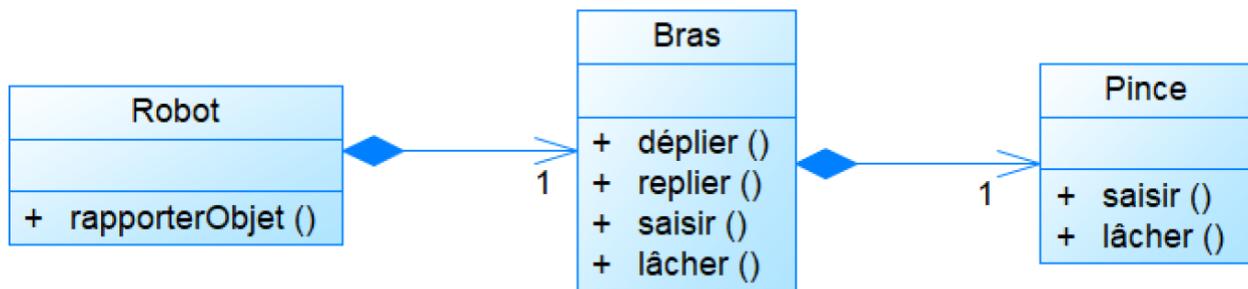
1. Une voiture a une marque, un modèle, un moteur. Modéliser la situation.
2. La marque appartient à une liste que vous choisirez (énumération). Modéliser la situation.
3. Le moteur a un nom et une puissance. Modéliser la situation.
4. La voiture a des roues (avant, arrière et de secours) et le moteur actionne 2 de ces roues. Modéliser la situation.
5. On souhaite préciser si le moteur actionne les roues avant ou les roues arrière. Modéliser la situation.
6. Faire un diagramme d'objets correspondant diagramme de classes précédent pour un objet de la classe voiture à traction avant.
7. Écrire le constructeur de la voiture sous la forme d'un diagramme de séquence. On peut aussi coder le constructeur (en plus du diagramme de séquence) si on préfère.

2 : Système de fichiers (d'après L. Audibert – Ellipses)

1. Un répertoire possède un nom et des droits en lecture, exécution et écriture. Modéliser la situation.
2. Un fichier a les mêmes caractéristiques. Mettre à jour le modèle précédent.
3. Un répertoire peut contenir des fichiers et des répertoires. Tous les fichiers sont dans un répertoire et un seul. Tous les répertoires sont dans un répertoire et un seul sauf le répertoire racine qui n'est dans aucun répertoire. Modélisez la situation.
4. Proposer un diagramme d'objets avec deux arborescences distinctes de hauteur 2 et 3, et avec des fichiers.
5. On peut effacer et renommer les fichiers et les répertoires. A partir d'un répertoire, on peut remonter à son parent, on peut lister son contenu, on peut se rendre dans un sous répertoire. Modélisez la situation.
6. On peut consulter ou modifier les droits d'un fichier ou d'un répertoire. Le principe est qu'on modifie les trois droits en même temps selon une syntaxe particulière. On veut aussi pouvoir accéder au nom d'un fichier ou d'un répertoire. Modélisez la situation.

3 : Le robot

1) Soit un robot modélisé comme suit :



Le robot qui dispose d'un bras articulé se terminant par une pince. Le robot peut rapporter des objets.

Faire le diagramme de séquence permettant de décrire cette méthode. On considère que pour rapporter un objet le robot doit déplier son bras, puis saisir l'objet avec sa pince, puis replier son bras et enfin lâcher l'objet.

Sur un logiciel, on commencera par reproduire le diagramme de classes pour pouvoir l'utiliser pour le diagramme de classe.

4 : Jeu de rôles et tournevis !

1) Soit un jeu vidéos avec des joueurs

Les joueurs ont un nom, une santé, une force et une expérience. Ces trois dernières caractéristiques sont des entiers positifs ou nuls. L'expérience vaut 0 quand le personnage débute.

On se dote d'un constructeur qui permet de donner un nom, une santé et une force au joueur.

On peut afficher les caractéristiques d'un joueur (son nom et la valeur de tous ses attributs).

Un joueur peut attaquer un autre joueur. Une attaque réduit la santé du personnage attaqué de 1. Quand la santé vaut 0, le joueur est mort. Une attaque augmente l'expérience de l'attaquant de 1.

1. Définir la classe, les attributs et les méthodes qui permettent de gérer cette situation.
2. Ecrire un *main* sous la forme d'un diagramme de séquence qui déclare deux joueurs et affiche leur caractéristiques. Ensuite le main fait attaquer le deuxième joueur par le premier. Afficher les nouvelles caractéristiques des joueurs. On peut aussi coder le main (en plus du diagramme de séquence) si on préfère.

2) On crée maintenant des ennemis

Les ennemis ont les mêmes caractéristiques que les joueurs mais ils n'ont pas d'expérience et ils ont une race (troll, elfe, sorcier, etc.), et une valeur qui est un entier positif. Quand un joueur attaque un ennemi, il gagne en expérience la valeur de l'ennemi. Quand un ennemi attaque un personnage, il lui fait perdre 1 point de santé.

3. Définir les classes, les attributs et les méthodes qui permettent de gérer cette situation.
4. Ecrire un *main* sous la forme d'un diagramme de séquence qui déclare deux joueurs et un ennemi, affiche leur caractéristiques. Ensuite le main fait attaquer l'ennemi par le premier joueur. Puis le main fait attaquer le deuxième joueur par l'ennemi. Afficher les nouvelles caractéristiques de l'ennemi et des joueurs. On peut aussi coder le main (en plus du diagramme de séquence) si on préfère.

3) Dans le jeu, les joueurs peuvent utiliser des outils

Ils peuvent trouver ou gagner des outils qui sont alors à leur disposition.

Un outil a un poids et un nom. On peut afficher les caractéristiques de l'outil.

Le tournevis est un outil qui a une forme particulière (cruciforme, plat, etc.) avec une taille donnée en entier.

Les tournevis permettent de visser et de dévisser des vis. Une vis est un objet qui à une forme et une taille comme le tournevis. Ainsi, un tournevis peut être adapté ou pas à une vis. Une vis peut être vissée ou dévissée (dans ce cas elle est disponible pour être vissée). Dans le jeu, les joueurs peuvent trouver ou gagner des vis qui sont alors à leur disposition.

Le joueur peut chercher une vis dans son stock qui correspond à un de ses tournevis et inversement chercher un tournevis qui correspond à une vis.

Il existe des tournevis électriques qui ont une batterie avec un niveau d'énergie. Pour visser ou dévisser, ces tournevis doivent avoir de la batterie.

5. Définir toutes les classes nécessaires pour gérer la situation.
6. Ecrire un *main* sous la forme d'un diagramme de séquence qui permet à un joueur de stocker un tournevis électrique et une vis qu'il a trouvé. Ensuite le joueur choisit une vis puis un tournevis électrique adapté pour cette vis et visse cette vis (on ne s'intéresse pas à savoir sur quoi il la visse !). On peut aussi coder le main (en plus du diagramme de séquence) si on préfère.

5 : L'application compteur

- 1) Modéliser une application qui simule un compteur : le compteur est initialisé à 0. Il peut être incrémenter de 1, décrémenter de 1 ou remis à 0. La valeur du compteur ne peut pas être négative. On modélisera aussi l'interface et un écran permettant d'utiliser ce compteur. Pour cet écran, on se donne deux méthodes : `traiterEvénement (unEvénement : Evenement)` qui sera capable de récupérer des clics souris et d'y associer des traitement et `afficherCompteur(Compteur)` ; et on se donne un type `Bouton` et une classe fenêtre dont hérite l'écran.
- 2) Modifier le modèle en ajoutant la possibilité de choisir la valeur de départ.
- 3) On souhaite faire évoluer le modèle tout en conservant l'existant et proposer un compteur dont l'incrément soit une autre valeur que 1 (2, 5, 10, etc.) fixée . Proposer un modèle.
- 4) On souhaite maintenant pouvoir utiliser un compteur d'entiers et un compteur de réels. On ne s'intéresse plus à l'application qui utilisera le compteur, mais on souhaite par contre que les compteurs offrent la possibilité d'être affichés.

6 : Les figures

- 1) Modéliser avec un diagramme de classes l'utilisation de figures graphiques : le carré, le rectangle, le cercle et un figure composé d'un carré dans un cercle.
Un cercle est défini par son centre qui est un point avec 2 coordonnées x et y.
Le rectangle est défini par son centre et la taille d'un côté horizontal et celle d'un côté vertical.
Le carré est défini par son point bas-gauche et la taille de son côté.
A noter qu'un carré est un rectangle particulier.
La figure composée d'un carré dans un cercle est telle que le centre du carré correspond au centre du cercle et que les sommets du carré sont sur le cercle.
- 2) On peut afficher les figures ou les effacer (les rendre invisible). On peut aussi afficher les points ou les effacer.
- 3) On peut récupérer les sommets des carrés et des rectangles
- 4) On peut déplacer les figures.
- 5) On peut changer la taille des figures. Pour cela, une méthode reçoit un coefficient multiplicateur. S'il est compris entre 0 et 1, la figure diminue de taille. S'il est supérieur à 1, la figure augmente de taille.
- 6) Faire un diagramme de séquence qui modélise le déplacement d'un cercle
- 7) Faire le diagramme de séquence qui modélise le déplacement d'un carré dans un cercle
- 8) Faire un diagramme de séquence qui modélise l'instanciation d'un carré dans un cercle. On ajoutera les constructeurs en conséquence.
- 9) Faire un diagramme de séquence qui modélise le changement de taille d'un carré dans un cercle

7 : Equation du second degré

On veut modéliser un diagramme de classe pour résoudre des équations du 1 et du second degré ($aX + b = 0$ et $aX^2 + bX + c = 0$).

On rappelle qu'une équation du premier degré peut avoir 1 solution ($-b/a$) ou 0 solution ou une infinité de solutions.

On rappelle qu'une équation du second degré peut avoir 2 solutions ($-b - \text{racine}(\text{delta}) / (2*a)$) et ($-b + \text{racine}(\text{delta}) / (2*a)$) ou 1 solution ($-b / (2*a)$) ou 0 solution ou une infinité de solutions.

On rappelle que $\text{delta} = b^2 - 4*a*c$.

On traitera le delta comme un objet à part.

L'objectif est de pouvoir récupérer les nombres de solution et la liste des solutions, quelle que soit l'équation ainsi que de pouvoir faire un affichage textuel basique. On souhaite aussi pouvoir récupérer le delta dans le cas d'une équation du second degré.

1. Faire le diagramme de classes.
2. Ecrire un *main* sous la forme d'un diagramme de séquence qui qui permette de créer une équation du premier degré et d'afficher les résultats. Idem avec une équation du second degré. Dans ce cas, on affichera aussi le delta.

8 : Tetris

Soit le jeu de Tétris suivant :

<http://www.gratuit-en-ligne.com/gratuit-jeux-en-ligne/jeu-tetris-gratuit-en-ligne.php>

1)

En regardant le fonctionnement du jeu, proposer un diagramme de classes modélisant le jeu. N'oubliez pas de mettre les méthodes nécessaires.

2)

Ecrire un algorithme de gestion événementiel qui gère le programme.

2 - Diagramme de classe métier

1 : La cinémathèque

Une cinémathèque veut mettre en place un système d'information concernant des films et leurs diffusions. La cinémathèque gère plusieurs sites dans différentes villes (un site par ville). Pour chaque film on connaît son titre, sa date de création, ses acteurs avec leur nom, leur prénom et leur nationalité, les réalisateurs avec les mêmes informations, la langue du film. On connaît aussi les salles dans lesquelles il a été projeté, avec l'heure et le jour de la projection, la version projetée (version originale ou version française). On connaît le nombre de places des salles, le type d'écran et le type de son, ainsi que le site correspondant, avec son nom, son adresse, son téléphone et le nombre de salles du cinéma.

Chaque responsable de site gère sa programmation et le bilan en terme de nombre de spectateurs. La programmation se fait film par film. L'enregistrement du nombre de spectateurs se fait jour par jour.

Le service « cinémathèque » gère la partie film de la BD.

Le service matériel crée et met à jour les données concernant les sites et les salles.

Tous les utilisateurs peuvent consulter les informations concernant les films, les programmations, les salles et les sites. Ils peuvent aussi consulter des informations sur la fréquentation des films. On propose un classement hebdomadaire, mensuel et annuel du nombre d'entrées par film, par réalisateur et par acteur.

1. Concevez le diagramme de classes métier

2 : L'association

Objectif : MEA et Diagramme des classes

Une association organise des chantiers d'été. Il y en a de 3 sortes : chantiers de réhabilitation, chantiers de construction ou chantiers archéologiques. Pour les chantiers de construction et de réhabilitation, on connaît le type de bâtiment. Pour les chantiers de réhabilitation, on connaît la date de création de l'ancien bâtiment et des informations sur l'ancien bâtiment. Pour les chantiers archéologiques, on connaît la période archéologique et la surface du chantier (qui peut être très grande). Les chantiers ont une date de début et une date de fin (qui peut être plusieurs années après la date de fin). Des équipes viennent sur le chantier. Une équipe est constituée pour une durée limitée (environ un mois pendant l'été). Elle est affectée à un chantier et un seul. Les équipes sont constituées de participants. Les participants aux chantiers sont des personnes. Leur participation à l'équipe peut être inférieure à la durée de vie de l'équipe elle-même. Les personnes ont un nom, un prénom, une adresse, un email et un téléphone.

L'association peut recevoir des dons de personnes. Les dons ont un montant et une date de don. Ils sont faits par une personne. Ils donnent lieu à un reçu fiscal envoyé à une certaine date.

L'association envoie des courriers aux personnes. Les courriers ont un titre, une date et un libellé.

L'association gère aussi des membres. Les membres payent des cotisations annuelles. Les cotisations ne donnent pas lieu à un reçu fiscal. On distingue donc deux types de dons : ceux avec reçu et les cotisations.

Les membres de l'association peuvent participer au conseil d'administration (CA). Le CA est renouvelé en général tous les ans. Les participants au CA ont une fonction particulière. Cette fonction peut changer en cours de CA. On conserve les dates de début et de fin de chaque fonction occupée. Chaque CA a un certain nombre de participants.

1. Concevez le diagramme de classes métier

3 : L'hôtel

Un hôtel est composé d'au moins deux chambres. Chaque chambre dispose d'une salle d'eau qui peut être une douche ou une salle de bain. L'hôtel héberge des personnes. Il peut employer du personnel et est dirigé par un des employés. L'hôtel a les caractéristiques suivantes : une adresse, le nombre de pièces, la catégorie. Une chambre est caractérisée par le nombre et le type de lits, le prix et le numéro.

L'hôtelier peut consulter la disponibilité des chambres, réserver une ou plusieurs chambres pour un client, annuler une réservation, facturer un séjour. La réservation d'une chambre se fait pour une certaine durée. On garde la trace des annulations.

Pour gérer le suivi de sa clientèle, l'hôtelier peut consulter l'historique des séjours de ses clients.

Le prix des chambres est calculé en fonction du type de chambre, de la durée et du nombre d'occupants.

L'hôtelier peut aussi faire le bilan de son exploitation bilan. Le bilan financier prend en compte le chiffre d'affaire et le calcul de la masse salariale. La masse salariale est calculé en sachant que le taux de charges patronales est de 80% et est constant pour chaque employé.

Un employé est caractérisé par son nom, son prénom, son sexe, sa date de naissance et son salaire brut annuel. Un client est caractérisé par son nom, son prénom et son adresse.

On peut créer un employé à partir de son nom, son prénom et sa date d'embauche.

On peut afficher toutes les caractéristiques d'un employé dont son âge et faire des mises à jours.

1. Faire le diagramme des classes.
2. Faire le diagramme de séquences de la réservation d'une chambre par un client. Pour réserver une chambre, il faudra d'abord consulter la disponibilité des chambres.

4 – Interfaces et design pattern

1 : Le téléviseur (Interface)

Modéliser un objet téléviseur en tant qu'objet utilisable par une application écrite dans un langage objet. Le téléviseur est décrit comme suit :

- Un téléviseur possède des haut-parleurs, un mécanisme de mise sous tension (alimentation), un mécanisme de changement de chaîne (tuner), un mécanisme de réglage du son par augmentation ou diminution (ampli), un écran associé à un mécanisme de réglage du contraste et de la couleur des images par augmentation ou diminution.
- Il affiche en permanence le niveau sonore, la chaîne et l'état allumé ou éteint.
- Il peut être déplacé n'importe où et fonctionner correctement si le courant et l'antenne ou le câble sont connectés
- Le téléviseur peut être utilisé avec ou sans télécommande.
- L'écran a certaines caractéristiques (largeur, hauteur, etc.). Les haut-parleurs ont certaines caractéristiques (puissance, etc.)

1 : Faire le diagramme de classes compact : les composants du téléviseur sont traités comme des attributs de type classe. On précisera les interfaces.

2 : Déployer le modèle précédent : les composants du téléviseur seront traités comme des classes à part avec leurs attributs et leurs méthodes.

3 : Le téléviseur peut être connecté à un magnétoscope externe pour permettre l'enregistrement d'émissions ou la lecture de films. Le téléviseur ou sa télécommande permet de choisir le mode tuner (en choisissant une chaîne) ou vidéo. L'enregistrement se fait automatiquement à partir de la chaîne courante quand le magnétoscope est mis en mode enregistrement. Mettre à jour le diagramme de classes.

4 : Ajouter les interfaces de chaque composant.

2 : Les canards (DPTLP)

On gère des canards. Tous les canards peuvent voler, cancaner, nager et on peut afficher une phrase qui les décrit.

On a plusieurs sortes de canards : des colverts et des mandarins, qui sont des vrais canards, qui volent et qui font « cancan », des canards en plastique qui ne savent pas voler et qui font « coin-coin », des leurres qui ne savent pas voler et qui sont silencieux.

On a aussi des prototypes de canards qui, par défaut, ne savent pas voler et font « cancan ».

On écrira un programme qui crée une liste avec tous les types de canards puis affiche ce qu'ils sont, les fait cancaner et voler.

Enfin, on modifiera le vol par défaut du prototype de canard pour le faire passer à un vol à réaction !

Modéliser le diagramme des classes permettant de gérer la situation.

Quel design pattern avez-vous intérêt à utiliser ?

3 : Dessiner et imprimer (DP)

On part d'un code dans lequel un client, la classe ServiceWeb utilise une classe, la classe DocHTML. La classe DocHTML fournit 3 méthodes : un setter, une méthode « dessine » et une méthode « imprime ».

On récupère une classe ComposantPDF. Dans ce composant, l'affichage se fait en trois étapes.

Intégrez le composant PDF dans le système et utilisez-le comme la classe DocHTML a été utilisée par le ServiceWeb. Pour cela vous devrez commencer par définir une interface en considérant que cette interface correspond directement à celle de la classe DocHTML. Il faut donc ensuite adapter l'interface de la classe ComposantPDF.

Modéliser le diagramme des classes permettant de gérer la situation.

Quel design pattern avez-vous intérêt à utiliser ?