

TE1 5,5 Poids 2 Moy. Theorie
TE2 5,2 1 5,4

TE Génie Logiciel - Rattrapage

Nom : MINDER VALENTIN

Date : 2 Juin 2015

42 points total

①	②	③	④	⑤	⑥	⑦	⑧	
2	5	2,5	1,5	10,5	4	8	1	34,5
2	5	2,5	1,5	11	8	8	4	42

→ 5,21

EXERCICE 1 – 2 POINTS

Méthodes XP

Si l'équipe de développement change en cours de route (départs, nouveaux engagements), doit-on revoir la vélocité?

Réponse à justifier

oui. la vélocité définit la charge de travail accomplissable par une équipe définie - elle change donc si cette équipe change.

EXERCICE 2 – 5 POINTS - 1.5 PAR RÉPONSE FAUSSE

Parmi les assertions suivantes, cochez celles reflètent correctement la méthode XP.

- ☒ Avec XP, une des toutes premières itérations consiste à mettre en place l'architecture du système.
- ☒ C'est le « manager », appuyé par le « coach », qui organise la répartition du travail entre les différents collaborateurs au sein de chaque itération
- ☒ Avec XP, les spécifications détaillées sont écrites par le client lui-même. Il les exprime en écrivant les tests « de recette » (tests fonctionnels).
- ☒ Tests de recette et tests unitaires doivent être dans leur ensemble exécutés de manière automatique
- ☒ Avec XP, seul le développeur est habilité à estimer le temps de réalisation d'un scénario

EXERCICE 3 – 2.5 POINTS

Méthode XP

Parmi les acteurs participant à un développement XP, dites quel est l'acteur principal qui accomplit chacune des activités énoncées ci-dessous :

- ☒ Rédaction des scénarios ? client
- ☒ Spécification des tests unitaires ? testeur
- ☒ Spécification des tests fonctionnels ? client
- ☒ Spécification du plan d'itération ? client
- ☒ Détermination de la charge de développement d'un scénario ? développeurs

EXERCICE 4 – 1.5 POINTS

Méthode XP

Travailler par binôme : Citer au moins 3 avantages cités dans le support de cours

- ☒ formation : 1 nouveau peut travailler avec en plus expérimenté
- ☒ contrôle mutuel au fur-et-à-mesure
- ☒ meilleure collaboration & stratégies d'équipe (tournus régulières)
- ☒ personne n'est "indispensable" au projet.

EXERCICE 5 – 11 POINTS – MODÉLISATION DE DOMAINE

Une compagnie d'aviation

Pour modéliser un système simplifié de réservations de vols pour une agence de voyages, nous partons d'un résumé de l'expertise métier donnée par les phrases suivantes:

1. Une compagnie aériennes – caractérisée par un nom - ^{affrète} propose différents vols.
2. Un vol, caractérisé par une date, est ouvert à la réservation et refermé sur ordre de la compagnie.
3. Un vol a un aéroport d'arrivée et de départ.
4. Les aéroports – caractérisés par un nom - peuvent desservir plusieurs villes – également caractérisées par un nom. Une ville peut être desservie par plusieurs aéroports. Pour une ville donnée, le taux de couverture de ses besoins que chacun de ses aéroports assure est enregistré ; son taux de couverture global doit atteindre 100%.
5. La compagnie possède des clients qui ont la possibilité d'opérer des réservations (une réservation ne concerne qu'un vol et qu'un seul passager)
6. Une réservation doit être confirmée dans les deux jours qui précèdent le vol par le client qui l'a opérée, sinon, la réservation n'est pas considérée comme valide.
7. Un client comporte un nom et une liste de liens permettant de le contacter: Il peut s'agir d'e-mails, de numéros de téléphone ou encore d'adresses.
8. Les « bons clients » bénéficient d'un rabais dont le montant est variable (minimum 10%).
9. Les passagers sont identifiés par un nom et une adresse

Remarques générales concernant les exercices de modélisation

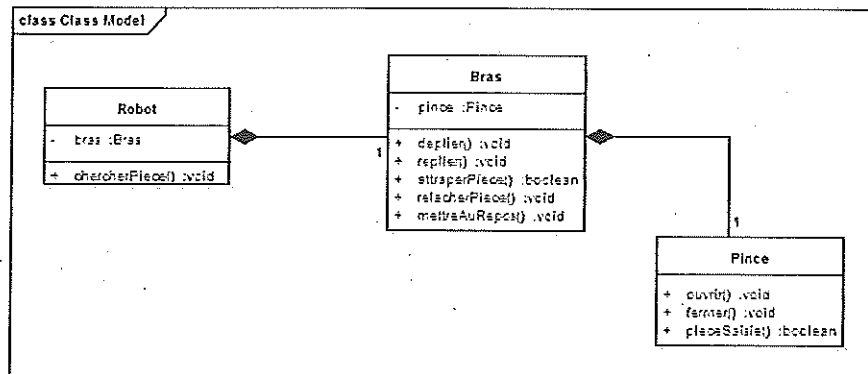
1. Ne pas rajouter d'informations qui ne se trouvent pas exprimées dans l'énoncé.
2. Toutefois, si l'énoncé vous paraît ambigu (cela peut arriver..), vous pouvez le compléter. Indiquez-le par une remarque.
3. Utiliser l'agrégation ou la composition partout où cela s'y prête. Accompagner votre schéma d'une justification (pourquoi agrégation plutôt que composition).
4. Nommez les associations simples
5. Utiliser la « Généralisation-Spécialisation » partout où cela s'y prête.
6. Utiliser des associations ternaires partout où cela s'y prête.
7. Indiquer la cardinalité de manière explicite (ne pas utiliser de valeurs par défaut)
8. Compléter votre schéma de modélisation par la liste des contraintes que la syntaxe des diagrammes ne permet pas d'exprimer.

EXERCICE 6 – 8 POINTS

DIAGRAMME DE SÉQUENCE

Un robot avec bras articulé

Le diagramme représenté ci-dessous modélise un robot disposant d'un bras articulé se terminant par une pince.



Représenter à l'aide d'un diagramme de séquence l'échange des messages entre les objets Robot, Bras, Pince et Log (voir plus bas) pour réaliser l'opération « **chercherPiece** ».

Le robot fonctionne de la manière suivante :

Pour chercher une pièce :

- le robot déploie son bras (la pince est d'abord ouverte avant de faire le mouvement),
- attrape la pièce (fermeture de la pince),
- replie son bras,
- et - si la pièce a été correctement attrapée -, relâche la pièce (ouverture de la pince),
- puis, se remet en position de repos (fermeture de la pince).

La méthode « `attraperPiece` » retourne un booléen indiquant le succès de l'opération, la fonction booléenne « `pieceSaisie` » retournant « `true` » si la pièce a correctement été saisie par la pince).

Pour des raisons d'aide au dépannage, l'opération « **chercherPiece** » commence par créer un objet de type « **Log** », - correspondant à un fichier -, mettant à disposition une méthode `ecriture(action: String)`. Cette méthode est invoquée en début d'exécution pour chacune des 2 méthodes '`deployer`' et '`replier`' de la classe Bras. Cette méthode écrit le texte correspondant ('`deployer`', '`replier`') dans le fichier Log. Si la pièce a été correctement saisie, cet objet est détruit quand l'opération « **chercherPiece** » se termine.

EXERCICE 7 – 8 POINTS

DIAGRAMME D'ACTIVITÉ

Gestion d'un hôtel

Reprendre le contexte de l'exercice de modélisation de domaine et dresser le diagramme d'activité correspondant au traitement du processus répondant à une demande de client.

Ce diagramme mettra en jeux 3 couloirs d'activité: le client, le conseiller, la gestion de l'hôtel.

Description du processus

Le client commence par formuler une demande. Cette dernière est analysée par le conseiller qui recherche alors une liste d'hôtels susceptibles de répondre à cette demande. Une proposition est alors envoyée au client.

Le client analyse la réponse. Il peut ne donner aucune suite, terminant ainsi le processus. Sinon, il communique son choix au conseiller.

Sur la base du choix opéré par le client, le conseiller envoie par e-mail un avis à l'hôtel qui a été sélectionné (pour signaler qu'une réservation sera opérée par le biais de l'agence) puis opère la réservation sur le site internet de l'hôtel.

Dès que l'avis et l'inscription, arrivant dans n'importe quel ordre, ont tous deux été reçus et enregistrés par le service administratif de l'hôtel, ce dernier envoie une confirmation par e-mail au conseiller.

Le conseiller, qui attendait cette confirmation, envoie alors une facture au client. Le conseiller attend alors, d'une part de recevoir une confirmation du paiement par le client, et, d'autre part, de recevoir la commission de l'hôtel.

Le client paye la facture. A L'HOTEL (confirmé!)

A la fin de chaque mois, l'hôtel envoie les commissions correspondantes aux agences qui leur ont envoyé des clients.

Le processus se termine une fois que le conseiller a reçu à la fois la confirmation du paiement et la commission de l'hôtel.

Note importante: toutes les communications d'informations sont opérées dans le diagramme de manière asynchrone.

EXERCICE 8 - 4 POINTS

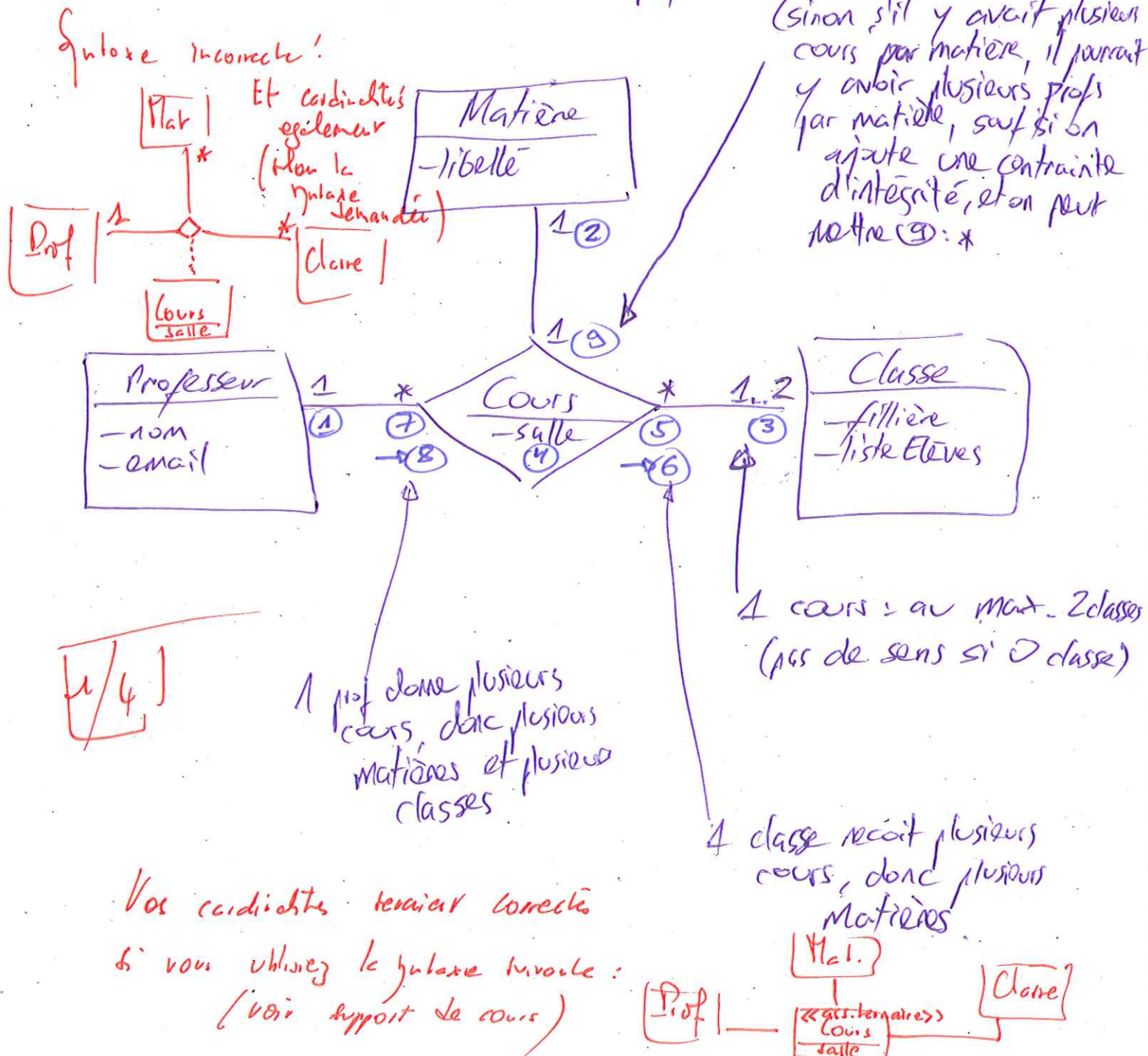
Donnez la représentation UML de l'association ternaire décrite ci-dessous (un « cours ») en utilisant la symbolique officielle (petit losange). Explicitez les contraintes à rajouter au modèle.

Compléments d'informations sur le modèle:

- ① Un cours est donné par un et un seul professeur et correspond à l'enseignement d'une matière.
- ② Un cours est donné à max. 2 classes simultanément (dans la même salle).
- ③ Une classe reçoit plusieurs cours et se voit enseigner plusieurs matières.
- ④ Un professeur peut enseigner plusieurs matières et enseigne à plusieurs classes
- ⑤ Une matière est enseignée par un professeur au maximum.

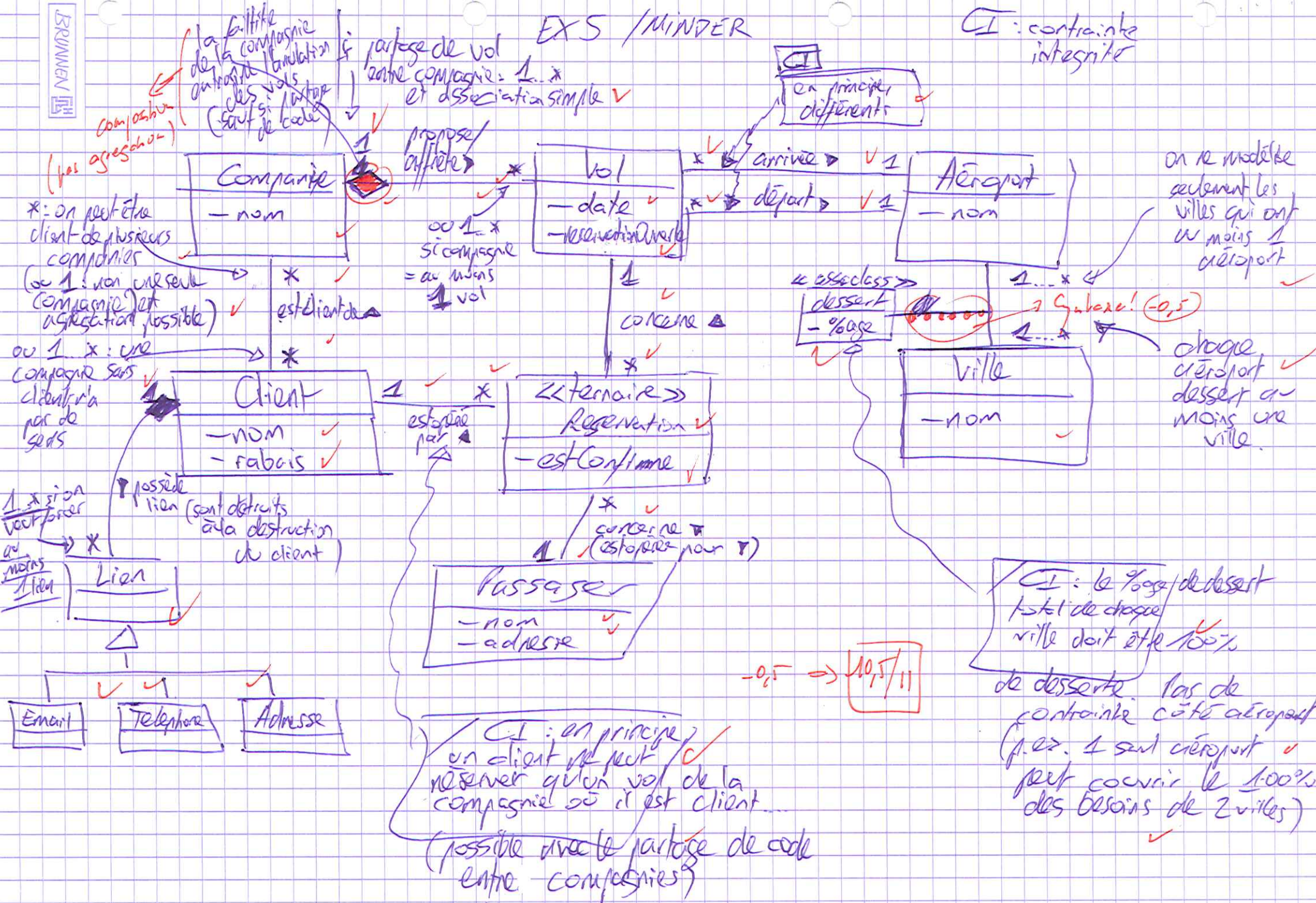
on a : 1 cours → 1 prof
et 1 matière → 1 prof

du coup :
1 matière → 1 cours
(sinon s'il y avait plusieurs cours par matière, il pourrait y avoir plusieurs profs par matière, sauf si on ajoute une contrainte d'intégrité, et on peut mettre ⑨ : *



EXS / MINDER

CI : contrainte intégrité



-4 => 4/8

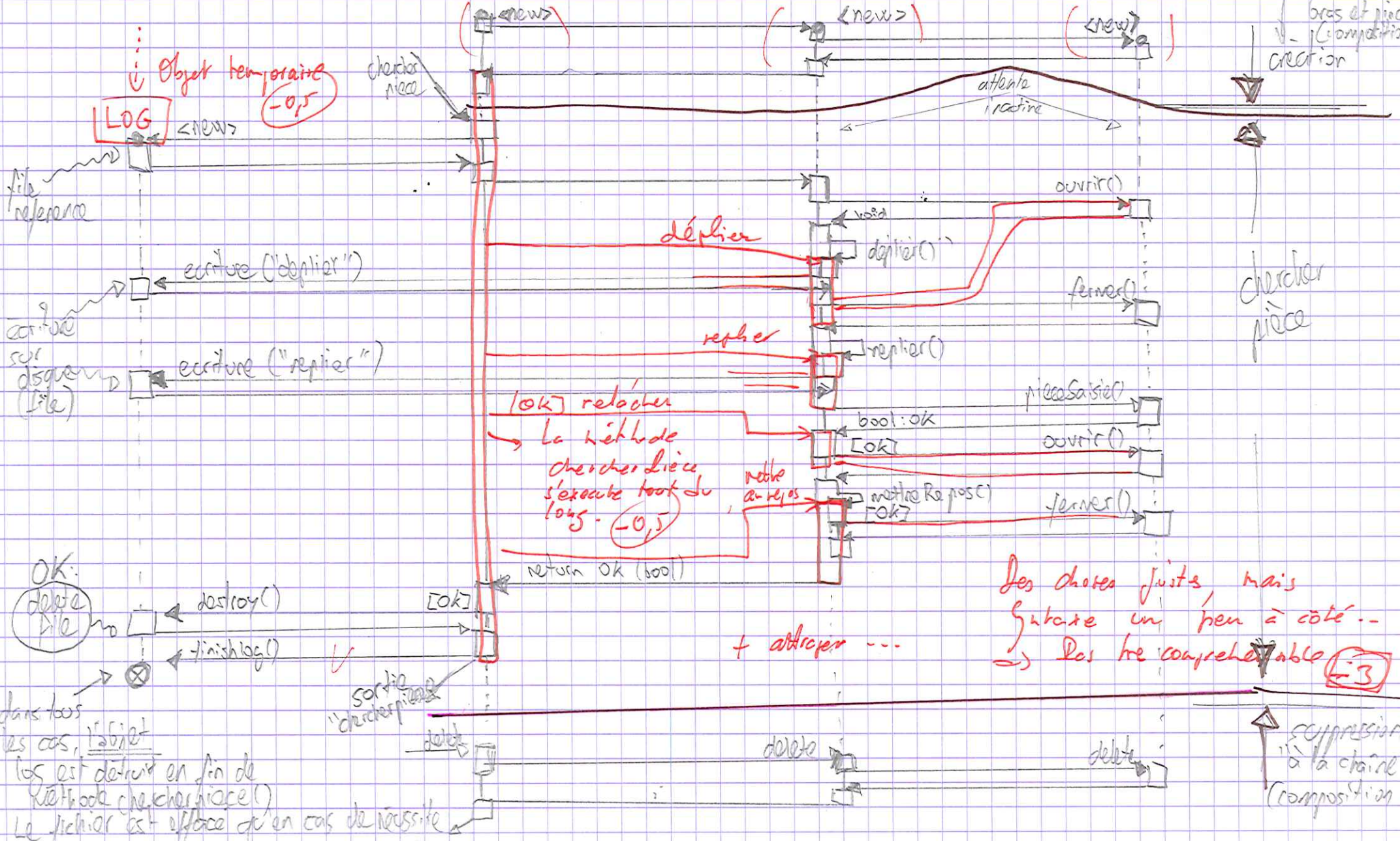
à la création,
le robot est
responsable
de la création
bras et pièce
(composition)

LOG

ROBOT

BRAS

PINCE



chercher pièce

des choses justes, mais
gubare un peu à côté...
=> pas très compréhensible (3)

OK: delete file

delete

finish log()

sortie "chercher pièce"

delete

dans tous les cas, l'objet log est détruit en fin de méthode chercher pièce()

Le fichier est effacé si en cas de réussite

