

# Examen Analyse et Conception Orientée Objet - ING1

Vendredi 8 juin 2012

Durée : 2h30

## Partie 1 : Questions de cours (3 points)

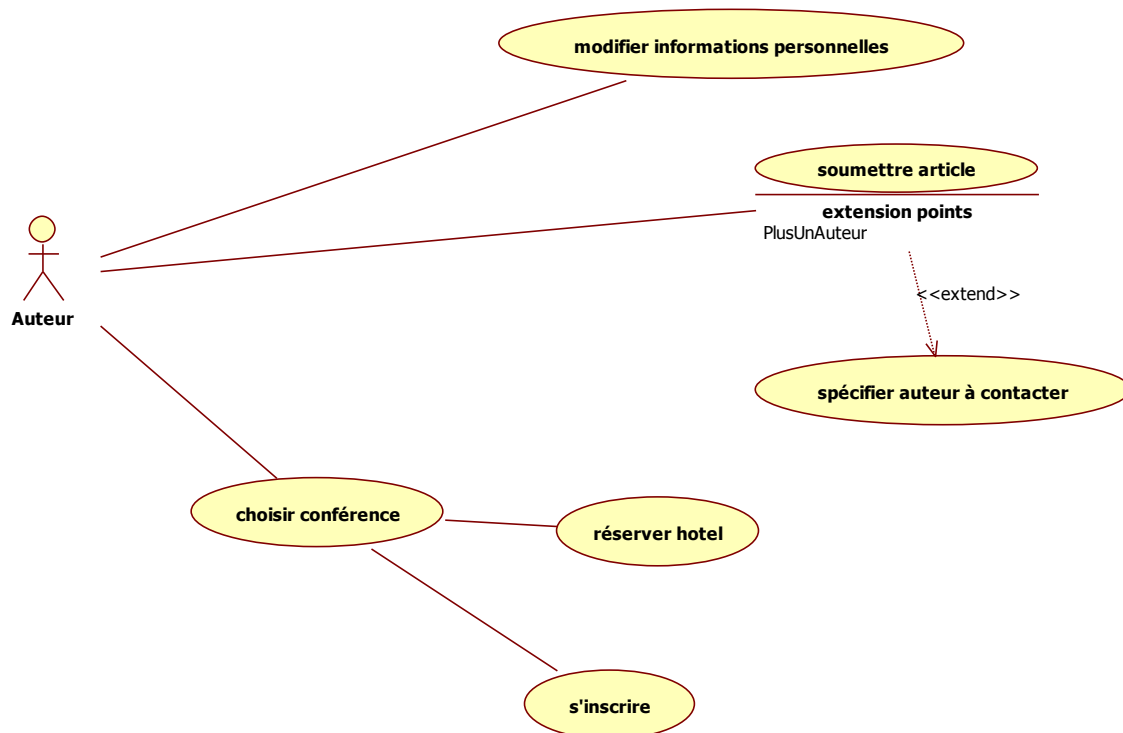
**Question 1)** Quelles sont les différences entre l'analyse et la conception ?

**Question 2)** Pourquoi faut-il plusieurs diagrammes d'UML différents pour un même projet ?

**Question 3)** Quelles sont les différences entre une interface et une classe abstraite ?

## Partie 2 : Diagramme de cas d'utilisation (3 points)

Dans un logiciel de gestion de conférences sur l'Internet, un auteur peut effectuer les opérations suivantes : modifier les informations personnelles, soumettre un article, réserver un hôtel et s'inscrire à une conférence. Quand un auteur veut soumettre un article, s'il y a plus d'un auteur, il faut spécifier qui sera l'auteur à contacter. Pour la réservation d'hôtel et l'inscription, il faut choisir une conférence dans la liste des conférences proposées. Voici un diagramme de cas d'utilisation pour ce système :



**Question 1)** Identifier toutes erreurs commises et proposer un diagramme corrigé.

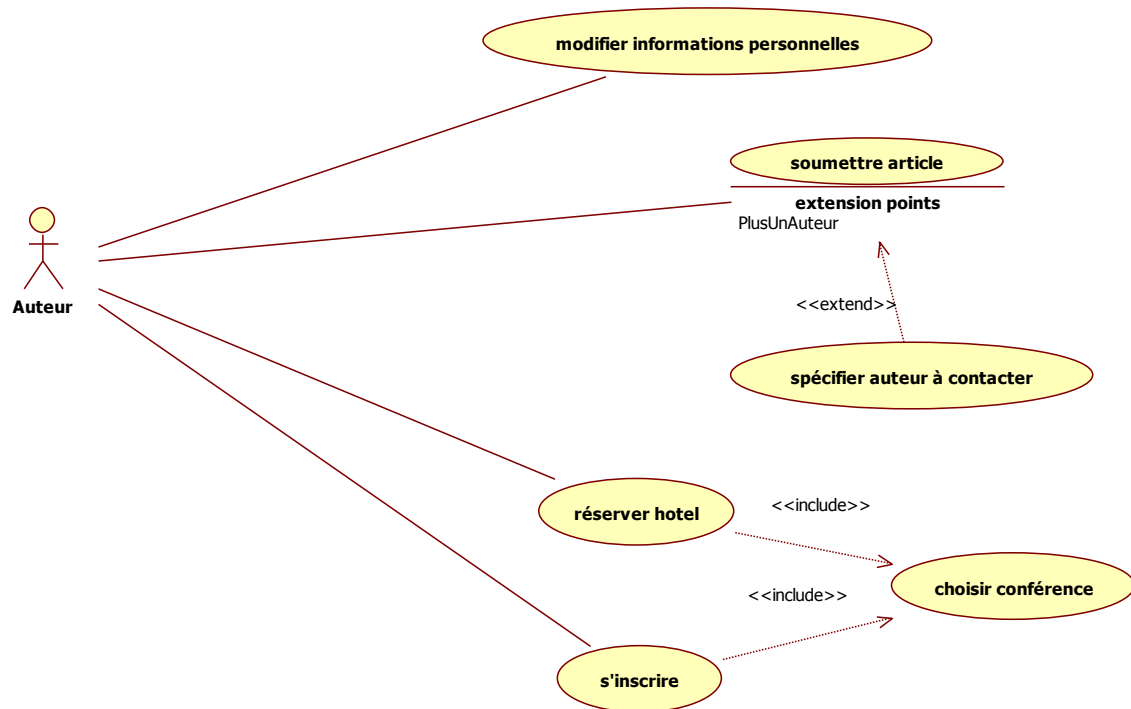
**Corrigé :**

Erreurs (1 point):

- Le sens de la flèche d'extension

- Les flèches entre "choisir une conférence" et "réserver hôtel", "s'inscrire" ne veulent rien dire.

Le diagramme corrigé :



### Partie 3 : Diagramme de classes avec OCL (7 points)

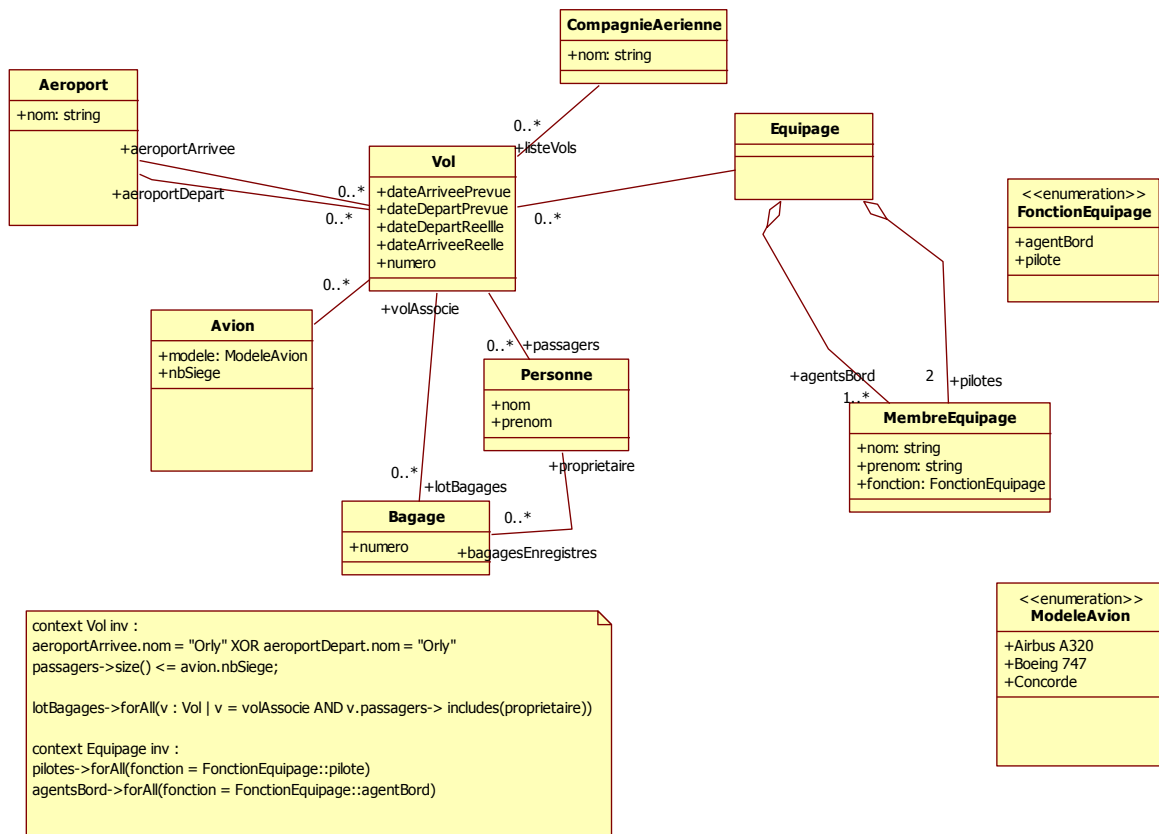
Nous modélisons un système de gestion des vols réguliers dans un aéroport international (dont le nom sera **Orly** dans cet exemple). D'après les interviews réalisées, on sait que :

- Les compagnies aériennes proposent différents vols réguliers
- Un vol régulier
  - a un aéroport de départ et un aéroport d'arrivée. Notre aéroport international (ex : Orly) sera donc forcément identifié comme celui de départ ou celui d'arrivée.
  - a un jour et une heure de départ, et un jour et une heure d'arrivée prévus.
  - peut avoir, malgré tout, du retard au départ ou à l'arrivée ; il peut aussi arriver en avance sur son horaire. Une propriété sera calculée selon les cas et affichera l'un des trois états suivants : « A l'heure », « En avance » ou « En retard ».
- L'équipage d'un avion est toujours constitué de deux pilotes et de plusieurs agents de bord. Chacune de ces personnes est identifiée par son nom, prénom et sa fonction. Un agent de bord ne peut être engagé comme pilote ou inversement.
- Un équipage et un avion sont affectés à un seul vol à la fois.
- Les agents de bord et pilotes peuvent appartenir à plusieurs équipages différents. Ceci dit, un membre ne peut appartenir à plus d'un équipage à la fois à un instant donné.
- Ces équipages doivent être opérationnels sur trois types d'avions : Airbus A320, Boeing 747 et Concorde.
- Les avions cités seront identifiés par la tour de contrôle de l'aéroport par leur modèle et le vol associé.

- Un type d'avion a un nombre fixe de sièges.
- Lors d'un vol donné, le nombre de passagers ne doit pas dépasser le nombre de sièges de l'avion affecté.
- Chaque vol lui est associé un ensemble de bagages enregistrés. Chaque bagage est forcément associé à un passager enregistré à ce vol.

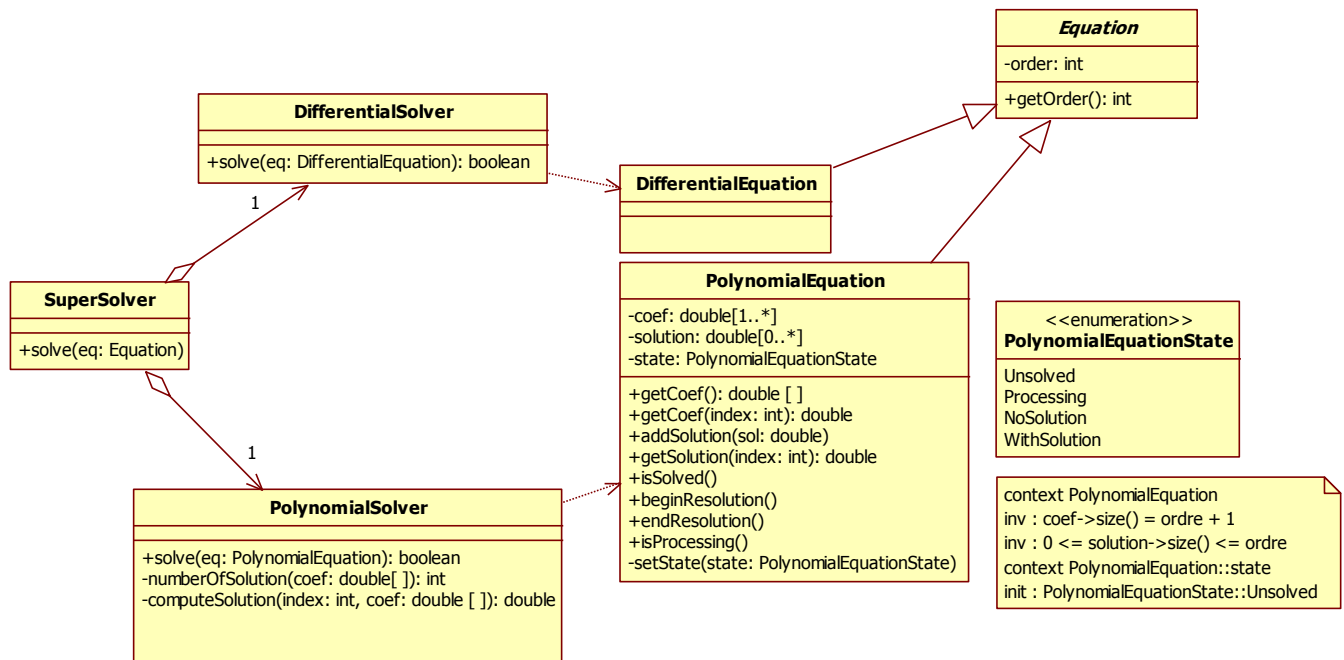
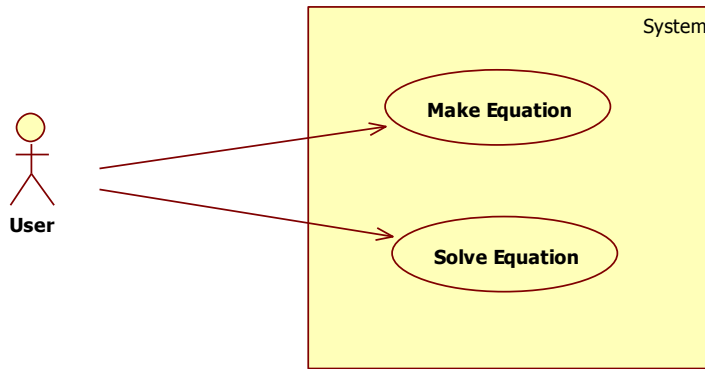
**Question 1)** A partir des éléments qui vous sont fournis ci-dessus, élaborer un diagramme de classes, les attributs et les associations/cardinalités nécessaires pour les décrire au mieux. Ajoutez toutes les contraintes OCL nécessaires pour compléter le diagramme.

**Corrigé :**



#### Partie 4 : Diagramme de séquence et diagramme d'états-transitions (7 points)

On considère un système de résolution d'équation dont on vous donne le diagramme des cas d'utilisation et le diagramme de classe.



Le scénario du cas d'utilisation Solve Equation est le suivant :

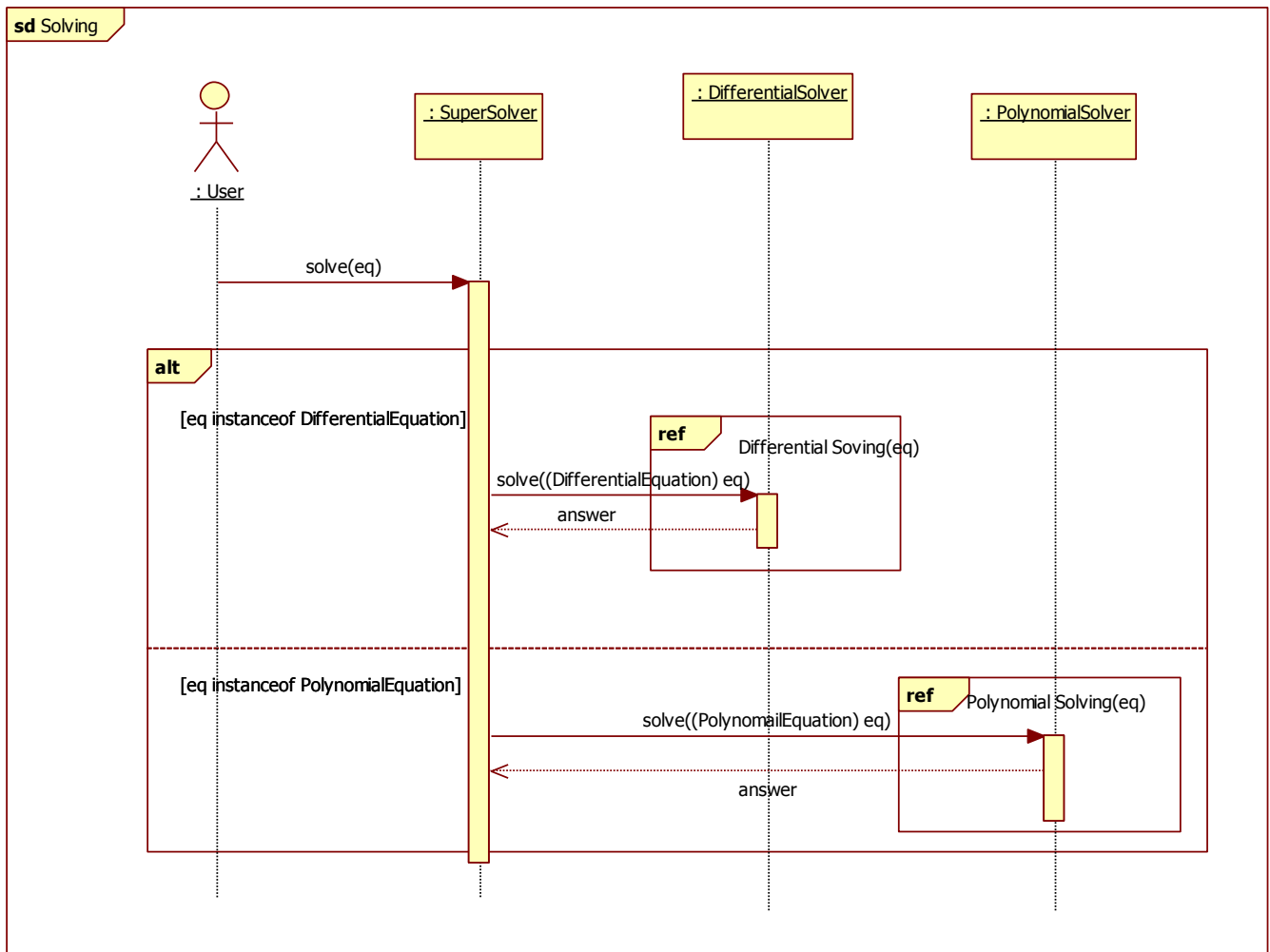
1. L'acteur user donne une équation à résoudre à un super solveur.
2. Suivant le type de l'équation le super solveur délègue la résolution au solveur polynomial ou au solveur différentiel.
3. Le scénario de la résolution différentielle n'est pas précisé. On le désignera uniquement par son nom : Differential Solving.
4. Pour le solveur polynomial :
  - a. Si l'équation est déjà résolue ou en cours de résolution, il s'arrête et renvoie vrai.
  - b. Il récupère l'ordre (i.e. degré) de l'équation polynomiale puis ses coefficients.
  - c. Il calcule le nombre de solutions N en fonction des valeurs des coefficients.
  - d. L'équation passe en mode Processing en signalant le début de sa résolution
  - e. Il calcule les N solutions en fonction des coefficients de l'équation.
  - f. Les solutions sont au fur et à mesure ajoutées à l'équation.
  - g. La résolution étant finie :
    - i. Le solveur signale à l'équation la fin de la résolution et cette dernière passe à l'état NoSolution ou WithSolution selon N.
    - ii. Le solveur renvoie vrai pour indiquer que la résolution est accomplie.

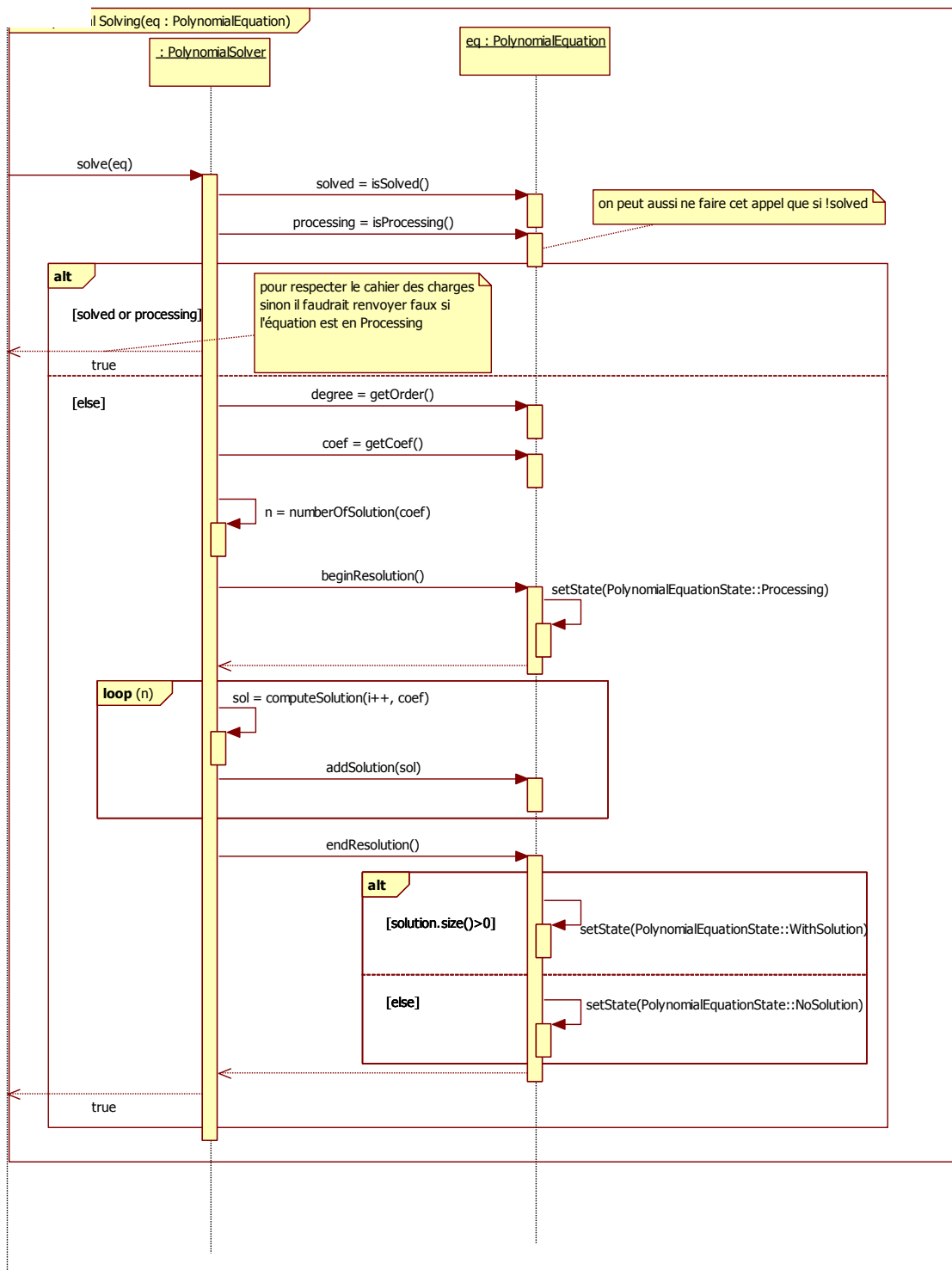
**Question 1)** Etablir le diagramme de séquence nommé Solving de ce scénario. Il est conseillé de découper ce diagramme en plusieurs sous-parties (une pour chaque solveur par exemple).

**Question 2)** Etablir le diagramme d'état d'un objet PolynomialEquation.

**Corrigé :**

**Question 1) 5 points**





Question 2) 2 points

PolynomialEquation State Diagram

