

## Environnement AGL Upper case PowerAMC

### Compétences Visées :

- ❖ Schématiser les diagrammes UML sous PowerAMC
- ❖ Générer le code (java comme exemple) à partir d'un diagramme de classes
- ❖ Générer une base de données relationnelle à partir du diagramme de classes

### Etude de cas

L'étude de cas prise en considération lors de ce TP concerne un système de réservation de vols pour une agence de voyages. Les interviews des experts métier ont permis de résumer leur connaissance du domaine sous la forme des phrases suivantes :

- ✓ Des compagnies aériennes proposent différents vols.
- ✓ Un client peut réserver un ou plusieurs vols.
- ✓ Une réservation concerne un seul vol et un seul passager.
- ✓ Une réservation peut être annulée ou confirmée.

## I. Introduction



PowerAMC™ est une solution graphique de modélisation qui prend en charge un large spectre de méthodologies et de notations standardisées (UML, Méthode Merise, etc.). En outre, il permet de procéder au reverse engineering et à la génération de code de façon automatique. Il fournit également une solution de référentiel d'entreprise évolutive grâce à ses de fonctionnalités robustes de gestion des projets de développement multiutilisateur (sécurité, gestion des versions, production des rapports, etc.)

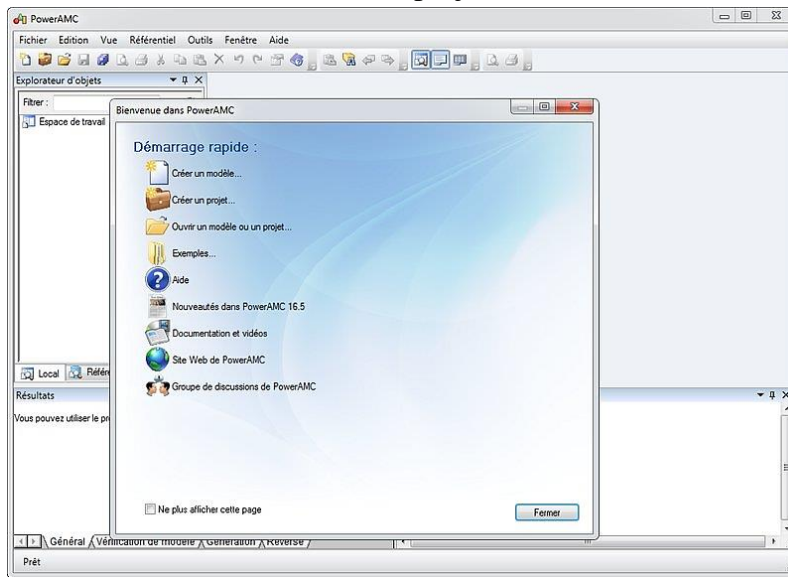
## II. Premier pas avec PowerAMC

Installer la version fournie (lors de la session du TP) de **PowerAMC** *ou bien* télécharger la nouvelle version (**PowerDesign**) à partir du site officiel [ici](#). Une documentation détaillée est disponible sur les liens suivants :

- Documentation en ligne : [ici](#)
- Aide-mémoire PowerAMC : [ici](#)

## 1. Création d'un nouveau Projet

PowerAMC, créer un nouveau projet : **Fichier** ➔ **Nouveau Projet**

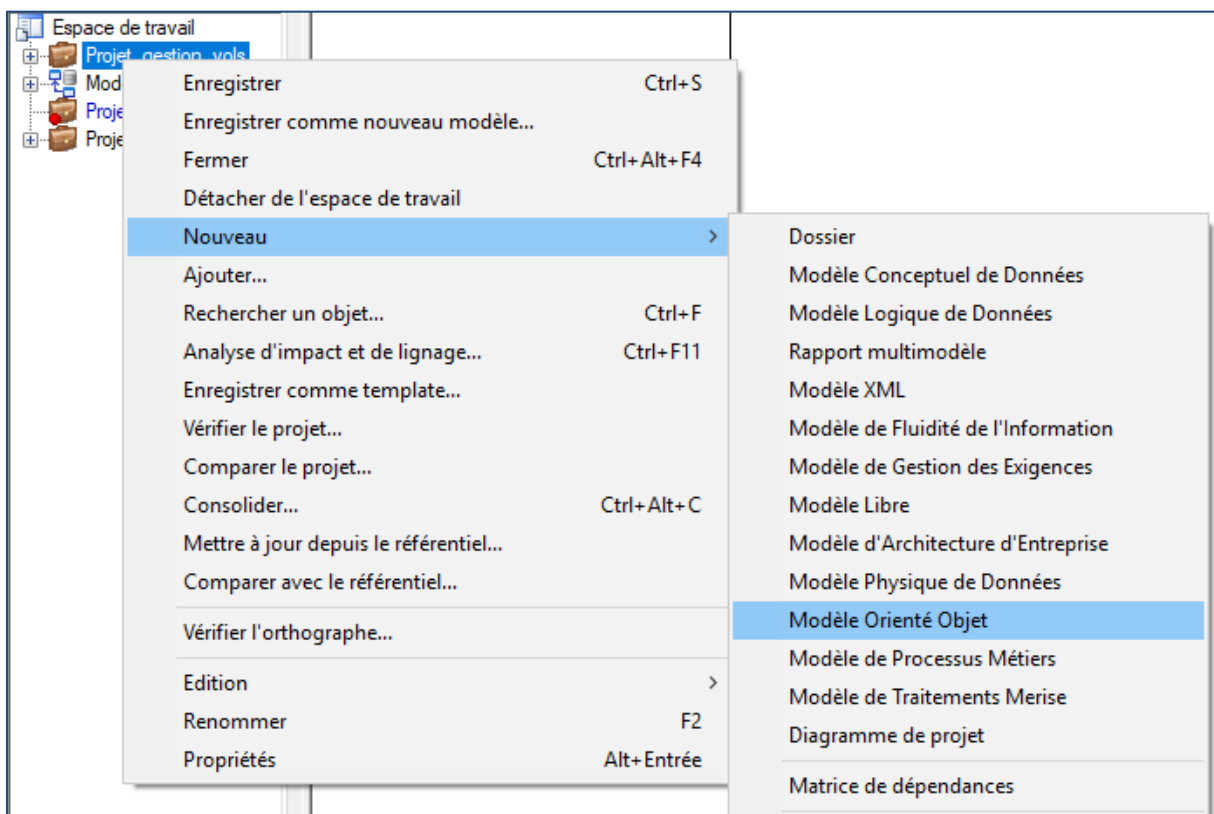


## 2. Création d'un nouveau Modèle orienté objet

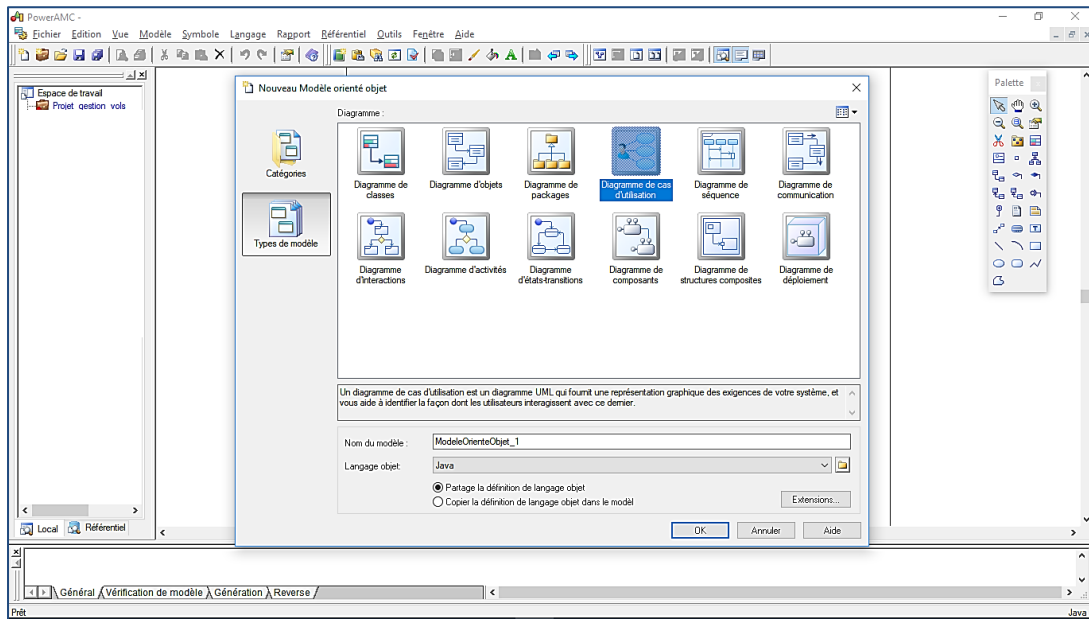
La modélisation du projet sera réalisée à base d'un modèle orientée objet. Un MOO est une structure qui fournit une description fidèle d'un système à l'aide des diagrammes UML.

Une documentation détaillée est disponible sur le lien suivant : [ici](#)

- Pour créer un nouveau **modèle orienté objet** au sein du projet :

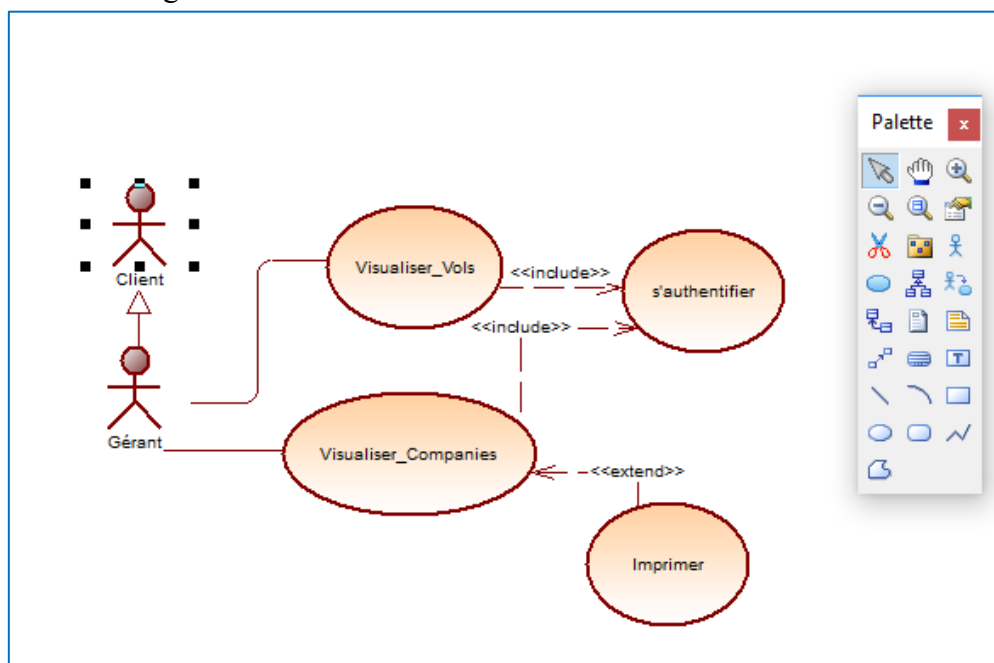


- La fenêtre suivante vous permettra de choisir les diagrammes que vous désirez ajoutés :



### 3. Réalisation d'un diagramme de cas d'utilisation

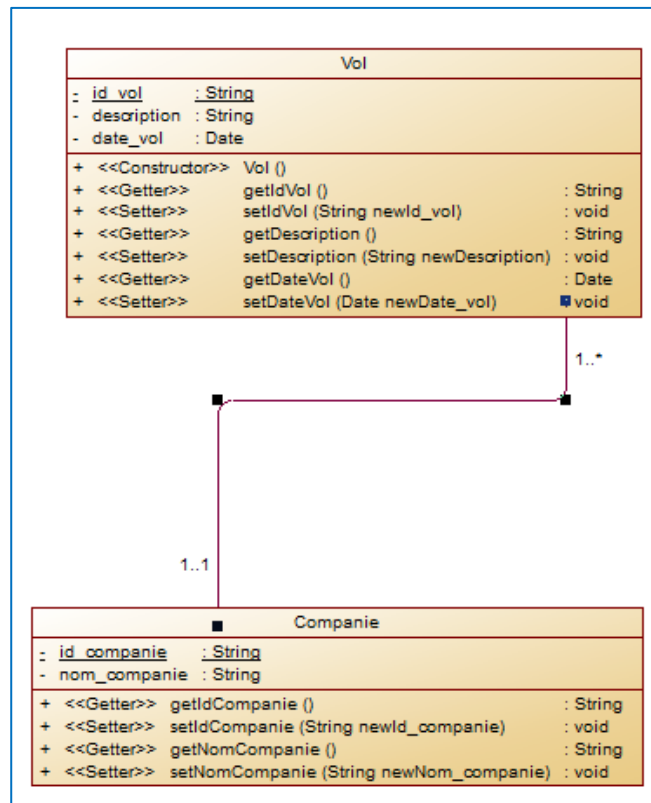
- Le diagramme de cas d'utilisation est construit en adoptant les étapes suivantes :
  - ✓ Identifier les acteurs externes concernés ;
  - ✓ décrire les fonctions du métier ;
  - ✓ établir les relations possibles entre les différents éléments identifiés (généralisation, inclusion, extension)
- Schématiser le diagramme de cas d'utilisation sous PowerAMC



### 4. Réalisation d'un diagramme de classes

- Le diagramme de classe est construit en adoptant les étapes :
  - ✓ identifier les concepts ;

- ✓ identifier et ajouter les associations et les attributs ;
  - ✓ organiser et simplifier le modèle (en éliminant les classes redondantes et en employant la notion de généralisation ou spécialisation) ;
  - ✓ le cas échéant, structurer les classes en paquetage selon les principes de cohérence et d'indépendance
- Schématiser le diagramme de classes sous PowerAMC.



## 5. Génération de code des classes

- Pour générer le code java des classes choisir l'option : **Fenêtre ➔ Générer du code java...**
- Générer le code java des classes identifiées.

## 6. Génération d'une base de données relationnelle

Généralement, le passage d'un diagramme de classes à un modèle relationnel est réalisé en ne gardant que les classes persistantes. Ce passage est possible grâce à une transformation des classes persistantes vers un modèle physique de données. Ceci peut être réalisé en respectant les points suivants :

- a) Chaque classe du diagramme devient une table en relationnel.
- b) A chaque classe du diagramme est associé un identifiant :
  - i. un identifiant à générer « classe\_id » ou bien une propriété de la classe.
  - ii. L'identifiant devient la clé primaire de la table correspondante.
  - iii. Toute relation dont une des cardinalités est de type (... ,1) provoque la génération d'une clé étrangère. Cela a pour conséquence la création d'une

colonne de nom « le\_rôle » dans la table. Chaque valeur de cette colonne doit faire référence à la clé primaire de l'autre table issue de l'association.

- c) Pour toutes autres cardinalités, la relation devient une nouvelle table ayant pour clé primaire le couple d'identifiant des tables issues de l'association.

❖ **Pour réaliser cette tâche :**

- générer d'abord un modèle physique de données du diagramme de classes en choisissant l'option : **Outils ➡ générer un Modèle Physique de Données.**
- Générer le code sql ; correspondant au SGBDR (Mysql par exemple) ; du modèle physique de données en choisissant l'option : **SGBD ➡ Générer la base de données...**

### 7. Reverse engineering d'une classe Java

Le reverse engineering est le processus qui consiste à extraire des connaissances ou des informations de conception d'un produit.

- Pour réaliser cette opération, sélectionnez **Langage ➡ Reverse engineering Java.**
- procéder au reverse engineering du fichier Java situé dans le répertoire associé à ce TP **Projet\_Gestion de vol\classes**. Le fichier Java contient une classe nommée Reservation.
- • Sélectionnez **Reservation.java**