

TP AGL avec Umbrello

1 Umbrello UML Modeler

Umbrello est un outil d'édition de diagrammes UML (Unified Modeling Language) développé sous **Linux**. Il permet de créer de manière simple et intuitive un très grand nombre de diagrammes UML, tel que le diagramme de classes. En plus de ses capacités très matures d'édition de diagrammes, Umbrello offre un grand nombre de fonctionnalités dont la plus intéressante est la génération de code source vers de nombreux langages (C++, Java, PERL, SQL, Ruby, Python, PHP, etc.).

Umbrello est disponible sur les machines du département. La version actuelle 2.8.4 est disponible en téléchargement libre (GNU General Public Licence) à cette adresse : <https://umbrello.kde.org/>. Pour les utilisateurs de Debian ou Ubuntu, le package est disponible dans les dépôts universes et peut être installé via Synaptic ou apt-get (`sudo apt-get install umbrello`).

Ressources :

- web : <https://umbrello.kde.org/>
- doc : <http://docs.kde.org/stable/en/kdesdk/umbrello/index.html> (documentation de la version 2.11)

Objectifs :

- Prise en main du logiciel Umbrello pour l'édition de diagrammes de classes,
- Observer/Maîtriser l'exportation du diagramme de classes en C++.

2 Sujet d'exemple

Des personnes (nom, prénom, date de naissance) sont de simples auditeurs ou des artistes. Les artistes ont sorti un ou plusieurs albums. éventuellement, un album est issu de la collaboration de plusieurs artistes. Chaque album est défini par son titre, le nombre de pistes, et sa catégorie. Chaque auditeur a une catégorie préférée et possède dans sa discothèque un ensemble d'albums.

Une solution est proposée sur le schéma 1.

3 Création d'une classe

Ouvrez Umbrello (raccourci Gnome : *Applications/Programmation/Umbrello*, commande : `/usr/bin/umbrello` ou `umbrello` dans un shell), sélectionnez le diagramme de classes et renommez le *catalogueCD*. Créez la classe *Personne*. Cliquez sur le bouton *Classe* pour créer une nouvelle classe dans le diagramme.

Ouvrez le menu d'édition de cette classe : *clic droit, propriétés*, et complétez cette classe :

- Via l'onglet *Attributs*, créez les 4 attributs de la classe *Personne* (`numPersonne`, `nomPersonne`, `prenomPersonne`, `dateNaissancepersonne`) en utilisant à chaque fois le type de donnée adéquat. Vérifiez que vos attributs sont privés ou protégés,
- A partir de l'onglet *Opérations*, créez une nouvelle méthode *Personne* avec les paramètres suffisant pour renseigner les attributs `nom`, `prenom` et `dateNaissance`. Comme cette méthode porte le nom de la classe, il s'agit d'un constructeur. Précisez-le (*Nom du stéréotype : constructor*). Vérifiez que cette méthode est publique.
- De retour sur la liste des opérations, sélectionnez l'opération *Personne* et ajoutez-lui le commentaire suivant : *Constructeur paramétré de la classe Personne (3 paramètres)*.

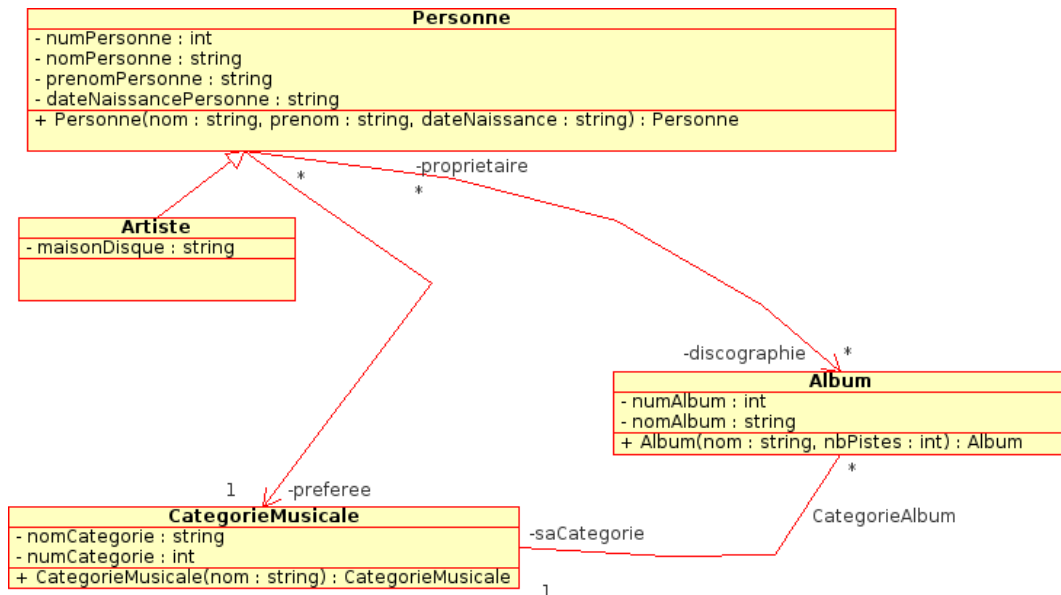


FIGURE 1 – Une solution au sujet d'exemple.

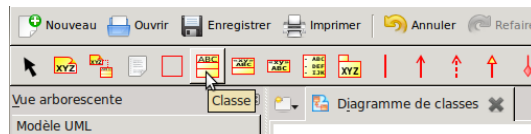


FIGURE 2 – Création d'une table.

- Ecrivez le code de méthode *Personne*.
- Dans l'onglet *Affichage*, décochez, si nécessaire, la case *publique seulement*.

Validez les propriétés de la classe *Personne*. Vous devriez obtenir une classe similaire à celle présente sur le schéma 1.

4 Exporter le diagramme de classes en C++

Nous allons maintenant exporter cette classe en C++ :

- Ouvrez l'assistant de génération de code : *Code* → *Assistant de génération de code...*
- Vérifiez que la classe *Personne* est bien sélectionnée pour exportation en C++ et cliquez sur *Suivant*,
- Dans l'onglet *Général*, sélectionnez le langage C++, et entrez un dossier d'exportation valide,
- Dans l'onglet *Mise en forme*, décochez les deux paramètres de verbosité des commentaires,
- Dans l'onglet *Options pour les langages de programmation*, sous-onglet *Général*, cochez toutes les cases,
- Cliquez sur *Suivant*, *Générer*, et *Terminer*.

Rendez-vous dans le dossier d'exportation que vous avez sélectionné à la troisième étape. Observez les fichiers créés. Lisez notamment le fichier **Personne.h**, qui contient la déclaration de votre classe *Personne*, commentée. Le constructeur par défaut et les accesseurs ont été implémentés. Votre constructeur paramétré est aussi déclaré, et documenté.

Générez le code en sélectionnant/désélectionnant les cases de l'onglet *Options pour les langages de programmation*, sous-onglet *Général* pour observer l'influence des options d'exportation sur le contenu du code généré.

5 Suite du diagramme de classes

- Comme vous avez créé la classe *Personne*, créez et éditez les classes *Artiste* (*maisonEdition*), *Album* (*numAlbum*, *nomAlbum*, *nbPiste*) et *CategorieMusicale* (*numCategorie*, *intituleCategorie*),

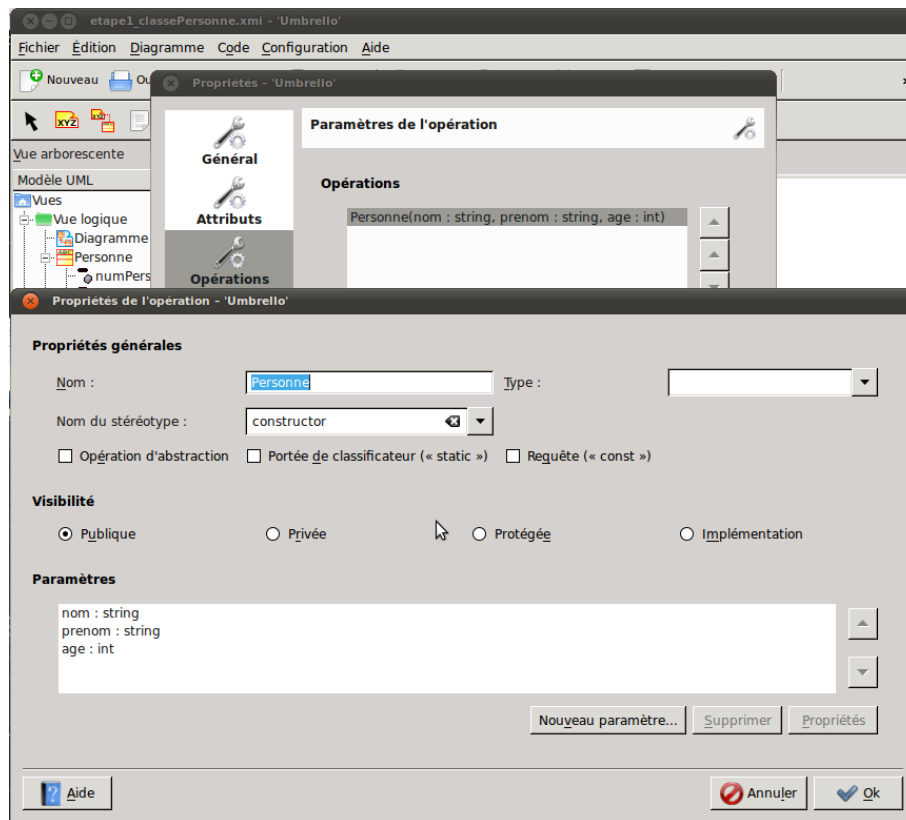


FIGURE 3 – Propriétés de la classe *Personne* et édition du constructeur paramétré.

- Créez les constructeurs paramétrés des classes *Album* et *CategorieMusicale*,
- Faites hériter *Artiste* de la classe *Personne* (bouton icône *Implémente*),
- Générez le code et vérifiez que la classe *Artiste* implémente bien la classe *Personne* (`class Artiste : public Personne`).

5.1 Création de l'association "Chaque personne a une catégorie préférée"

- Créez maintenant l'association entre *Personne* et *CategorieMusicale* pour spécifier la catégorie préférée de chaque auditeur (bouton *Association*, *Glisser/Déplacer* entre les deux classes),
- Ouvrez les propriétés de cette association (*clic droit, propriétés*),

Dans l'onglet *Rôles* :

- Ne donnez aucun nom au rôle de *Personne*, et vérifiez que la multiplicité est `*`,
- Donnez le nom *préféré* au rôle de *CategorieMusicale*, et vérifiez que la multiplicité est `1`,
- Ajoutez une documentation à ce rôle : **Détermine la catégorie musicale préférée de l'auditeur**,
- Choisissez une visibilité *privée* et la propriété de modification *modifiable* pour chaque rôle.

Exportez le code C++ et regardez le fichier `Personne.h`. Un attribut `Categorie* préférée` a été ajouté à la classe. Notez qu'il porte le nom du rôle, est documenté et que ses accesseurs sont implémentés.

Comme vous avez défini l'association *Chaque personne a une catégorie musicale préférée*, définissez l'association traduisant que *Chaque album est d'une catégorie musicale donnée*. Vérifiez l'exportation en C++. Un attribut `Categorie* <nom_role>` doit apparaître dans la classe *Album*.

5.2 Création de l'association entre *Artiste* et *Album*

Créez une association entre les classes *Artiste* et *Album* traduisant que *Chaque artiste a créé des albums*. Dans les propriétés des rôles de cette association, donnez les noms de rôle *oeuvres* au rôle de la classe *Album* et *concepteurs* au rôle de la classe *Artiste* (visibilités privées). Les multiplicités sont `(0..* - 0..*)`. Appliquez, générez le code C++ et regardez le contenu de `Artiste.h` et `Album.h`.

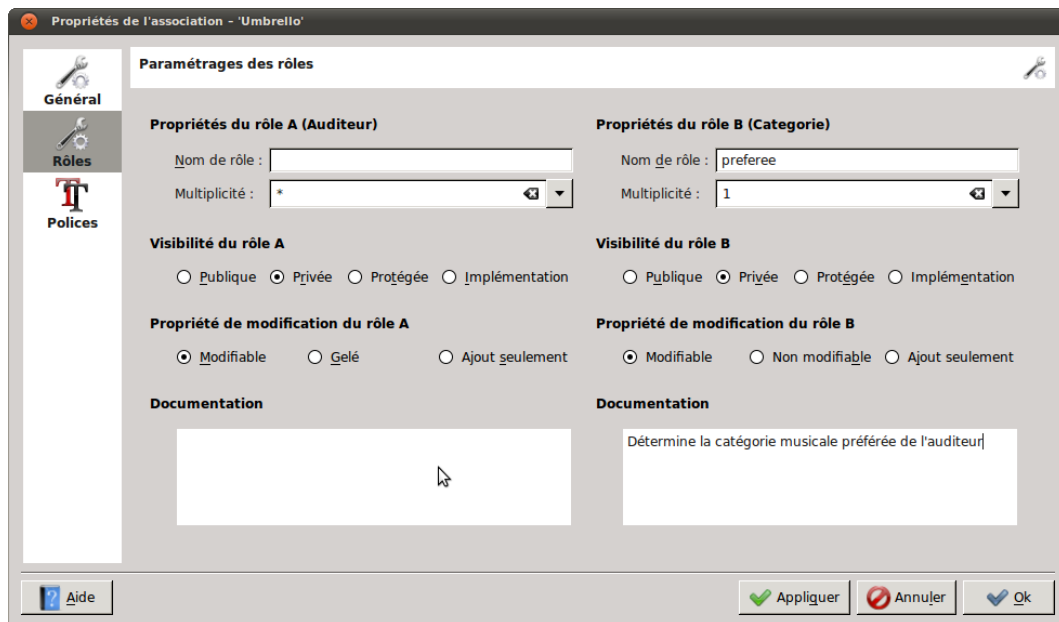


FIGURE 4 – édition des propriétés d’une association.

Vous pouvez observer qu’une liste d’artistes concepteurs apparaît comme attribut de la classe *Album*. De même, une liste d’albums *oeuvres* apparaît comme attribut de la classe *Artiste*. Rappelez-vous que **si vous donnez un nom de rôle d’un côté (ou de l’autre, ou des deux côtés) d’une association, il se traduit par un attribut de même nom dans la classe.**

Pour limiter l’exportation dans un seul sens, il faut utiliser la notion de visibilité (uni-association). Utilisez cette notion pour définir l’association traduisant le fait que chaque *Personne* possède une liste d’albums dans sa discothèque (c.f. schéma 1). Vérifiez en C++ qu’un attribut *discographie* est bien créé dans la classe *Personne* ; et qu’il n’y a pas d’attribut *propriétaire* dans la classe *Album*.

6 Rétro-conception : essai d’importation de code existant

Le but de l’exercice est d’examiner les capacités d’Umbrello à engendrer un diagramme de classe pour un projet déjà existant. Ouvrez le répertoire *Train* dans *Bibliothèque/S2-UML* et analysez le code fourni.

6.1 Analyse de code

Identifiez rapidement les classes et leurs relations (composition ? agrégation ? association ? quelles multiplicités ?), ainsi que les méthodes et leurs rôles.

6.2 Importation de code

À l’aide de l’assistant d’importation de code d’Umbrello, importez l’ensemble des fichiers d’en-tête du projet (dans un diagramme vierge!).

Que constatez-vous ? Comment afficher les classes ? Quelles sont leurs relations ?

6.3 Rétro-conception

Corrigez les erreurs d’importation en vérifiant les rôles, les multiplicités...

7 A rendre

A rendre à votre chargé de TD pour **le vendredi 27 mars 2015** : le diagramme de classes de la société Carloc (TD4) qui vous aurez complété (terminé), réalisé avec Umbrello.