Introduction à la POO
Classes, attributs, méthodes, instanciation,...

Manuel Munier

Version 10 septembre 2024 (14:06)

Objectifs

Pour ce second TP nous allons reprendre le même énoncé que pour le premier TP, mais en le programmant cette fois-ci en version OO, c'est-à-dire en écrivant des classes avec des attributs, des constructeurs et des méthodes, puis en instanciant des objets à partir de ces classes.

Consignes de programmation

Voici quelques consignes (de bon sens) pour prendre de bonnes habitudes en programmation OO :

- Limiter les attributs d'une classe au strict nécessaire ; comme en BdD, il n'y a aucun intérêt à stocker des résultats de calculs, sauf à introduire inutilement de la redondance d'information (i.e. une même information représentée plusieurs fois sous plusieurs formes différentes, avec des risques d'incohérence)
- Par défaut, mettez tous vos attributs en `private` afin de les rendre inaccessibles de l'extérieur de l'objet (→ abstraction des données).
- Penser à commenter vos codes sources : rôle d'une classe, fonctionnalité des méthodes, etc.
- Penser "modularité"...

Travail à réaliser

Dans cet exercice on se propose de programmer en Python un ensemble de classes nécessaire à la réalisation d'une application de gestion des étudiants et de leurs notes. Contrairement au premier TP où nous nous étions limités à des structures de données "classiques" (tableaux, tuples, listes, etc.), nous allons ici développer en **O**rienté **O**bjet, c'est-à-dire en utilisant des objets, classes et méthodes.

Voici dans les grandes lignes les principaux points à développer dans ce second TP :

1. Écrire la classe `Etudiant` avec des attributs permettant de représenter son nom, son prénom, la liste de ses notes avec les coefficients de chaque note. Cette classe devra disposer des méthodes suivantes :
   (a) une méthode `ajouterNote(...)` qui permet d'insérer une nouvelle note avec son coefficient à l'étudiant "sur lequel elle s'exécute"
   (b) une méthode `nbNotes(...)` qui retourne le nombre de notes de l'étudiant
   (c) une méthode `moyenne(...)` qui calcule la moyenne de l'étudiant

2. Écrire la classe `Promotion` avec des attributs permettant de son intitulé et sa liste d'étudiants (cf. des objets de la classe définie ci-dessus). Cette classe devra disposer des méthodes suivantes :
   (a) une méthode `ajouterEtudiant(...)` qui permet d'insérer un nouvel étudiant dans la promotion
   (b) une méthode `nbEtudiants(...)` qui retourne le nombre d'étudiants de la promotion

(c) une méthode `moyenne(...)` qui calcule la moyenne de la promotion, en excluant bien évidemment les étudiants qui ne disposeraient encore d'aucune note.

3. Plus, bien évidemment, des programmes de test en tout genre pour instancier des objets à partir des classes écrites ci-dessus, invoquer des méthodes sur ces objets, des fonctions diverses et variées pour déboguer (ex : affichage), etc.

Document rédigé en LaTeX sous Linux.