

GESTION D'UN ÉTABLISSEMENT SCOLAIRE « PARTIE 1 »

On voudrait réaliser une application orientée objet permettant de gérer les affectations des cours aux enseignants et aux étudiants d'un établissement scolaire, à l'aide des classes suivantes :

Une classe appelée **Personne** contient :

- 3 variables d'instance privées : nom, date_naissance (de type **DateTime**) et adresse (de type **string**), ayant les valeurs par défaut "inconnu", la date d'aujourd'hui moins 18 ans, et "inconnue" respectivement;
- 4 constructeurs : 1 constructeur sans arguments **Personne()**, 1 constructeur qui initialise le nom **Personne(string nom)**, 1 constructeur à 3 arguments **Personne(string nom, DateTime date_naissance, string adresse)**, et le constructeur de copie **Personne(Personne p)**;
- 3 accesseurs;
- 1 méthode ToString() qui retourne les informations sur une personne. Exemple :

« Nom : AYADI Ali. Né le : 10/10/1995. Adresse : 64, Rue Ain Boufares, Ouazzane. ».

Une classe appelée **Cours** contient :

- 3 variables d'instance privées : code_cours, intitulé_cours (de type **string**) et coefficient (de type **int**), ayant les valeurs par défaut "C00", "sans intitulé" et 1 respectivement;
- 2 constructeurs : 1 constructeur sans arguments **Cours()**, et 1 constructeur à 3 arguments **Cours(string code_cours, string intitulé_cours, int coefficient)**;
- 3 accesseurs;
- 1 méthode ToString() qui retourne le code et l'intitulé du cours uniquement. Exemple : « C11 : Programmation orientée objet. ».

Une classe appelée **Enseignant** hérite de la classe **Personne**, et contient :

- 2 variables d'instance privées : Matricule (de type **string**), et cours (de type List<**Cours**>) contenant la liste des cours enseignés par l'enseignant. Le matricule est généré automatiquement en commençant par 1. Toutefois le matricule doit toujours avoir 5 caractères, par exemple «00001»;
- 7 constructeurs :
 - 1 constructeur sans arguments **Enseignant()**;
 - 1 constructeur qui initialise le nom **Enseignant(string nom)**;

2 constructeurs qui initialisent les informations sur la personne :
`Enseignant(string nom, DateTime date_naissance, string adresse)`
et `Enseignant(Personne p);`

3 constructeurs d'initialisation :

`Enseignant(List<Cours>cours), Enseignant(Personne p, List<Cours>cours)` et `Enseignant(string nom, DateTime date_naissance, string adresse, List<Cours> cours);`

- 1 accesseur pour cours et 1 accesseur en lecture seule pour matricule;
- 1 méthode `ToString()` qui retourne les informations sur un enseignant, en respectant la forme donnée par l'exemple suivant :

« Matricule : 00001. Nom : MOULAH Ahmed. Cours enseignés : C01 :».

Une classe appelée `Etudiant` hérite de la classe `Personne`, et contient :

- 3 variables d'instance privées : `code_étudiant` (de type `string`), `cours` (de type `List<Cours>`) contenant la liste des cours suivis par l'étudiant, et `notes` (de type `double[]`) contenant la liste des notes obtenues dans les différents cours
- 3 constructeurs :
 - 1 constructeur sans arguments `Etudiant();`
 - 1 constructeur qui initialise le nom `Etudiant(int code, string nom, DateTime date_naissance, string adresse);`
 - 3 constructeurs qui initialisent la liste des cours :
`Etudiant(string nom, DateTime date_naissance, string adresse, List<Cours>cours, double []notes)`
- 1 méthode `Moyenne()` pour calculer la moyenne des notes de l'étudiant.
- 3 accesseurs pour cours et notes, `code_étudiant`, et 1 accesseur en
- 1 méthode `ToString()` qui retourne les informations sur un étudiant, en respectant la forme donnée par l'exemple suivant :

« Code : 199510250001. Nom : AYADI Ali. Moyenne : 14,26. ».

Travail à réaliser :

- Écrire les quatre classes `Personne`, `Cours`, `Enseignant` et `Etudiant`.
- Créer un jeu de tests.