

Nom: _____ Prénom: _____

Université Pierre et Marie Curie
TME SOLO - UE: Eléments de programmation orienté objet avec Java (2I002)

Créer une hiérarchie de classes pour représenter les étudiants d'une université. Il y a 3 types d'étudiants : ceux inscrits en Licence, ceux inscrits en Master, et ceux inscrits en Doctorat. Les étudiants peuvent s'inscrire à plusieurs UEs qui correspondent à leur niveau. Voici les descriptions de chaque classe.

1 (1.0 point) : La classe UE contiendra les variables et méthodes suivantes :

- code : valeur entière. Ce code ne peut pas changer de valeur, et doit être un nombre qui commence à un. La suite de codes doit être séquentielle.
- description : une chaîne de caractères avec la description de l'UE.
- niveau : valeur entière. Trois valeurs sont acceptées, 1 pour licence, 2 pour master et 3 pour doctorat. Créez de constante pour faciliter la réutilisation du code.
- notes : un tableau d'entiers avec trois positions.
- un constructeur à deux paramètres pour initialiser description et niveau. Le code doit être initialisé automatiquement.
- une méthode toString qui retourne « Code UE = » code « Description UE = » description « Niveau = » [licence ou master ou doctorat]
- méthodes get et set pour tous les attributs.

2 (2.0 point) : La classe Etudiant contiendra les variables et méthodes suivantes :

- numero : valeur entière.
- numCourrantUE : valeur entière, qui indiquera à combien d'UEs l'étudiant est inscrit jusqu'au présent.
- UEs : un tableau de UEs.
- constructeur qui prend en paramètre deux valeurs entières, le numéro de l'étudiant et combien d'UEs il/elle a le droit de s'inscrire au total.
- une méthode toString qui retourne « Etudiant Numero » numero
- une méthode afficheListeUEs qui affichera toutes les UEs d'un étudiant. Pour cela il faut faire appeler à la méthode toString de la classe UE.
- méthodes get et set pour tous les attributs.

3 (3.0 point) : La classe EtLicence hérite de la classe Etudiant, elle contiendra les variables et méthodes suivantes :

- parcours : chaîne de caractères.
- Un premier constructeur à trois paramètres pour initialiser tous les attributs.
- Un deuxième constructeur qui prend en paramètre le numéro de l'étudiant et initialise les autres attributs avec des valeurs par défaut (zéro pour le chiffre et vide pour les chaînes de caractères). Ce constructeur doit appeler le constructeur à trois paramètres.

- une méthode toString qui retourne « Etudiant Numero » numero « Licence, parcours=» parcours.
- une méthode ajouterUE qui prend en paramètre l'UE où l'étudiant veut s'inscrire. Les étudiants de licence ont le droit de s'inscrire seulement aux UEs de licence. Si l'étudiant a le droit de s'inscrire, l'UE est ajoutée à sa liste d'UEs et la méthode renvoie true, sinon rien n'est ajouté et false est renvoyé.
- méthodes get et set pour tous les attributs.

4 (3.0 point) : La classe EtMaster hérite de la classe Etudiant, elle contiendra les variables et méthodes suivantes :

- spécialité: chaîne de caractères.
- Un premier constructeur à trois paramètres pour initialiser tous les attributs.
- Un deuxième constructeur qui prend en paramètre le numéro de étudiant et initialise les autres attributs avec des valeurs par défaut (zéro pour les chiffres et vide pour les chaînes de caractères). Ce constructeur doit appeler le constructeur à trois paramètres.
- une méthode toString qui retourne « Etudiant Numero » numero « Master, spécialité =» spécialité.
- une méthode ajouterUE qui prend en paramètre l'UE où l'étudiant veut s'inscrire. Les étudiants de master ont le droit de s'inscrire seulement aux UEs de master. Si l'étudiant a le droit de s'inscrire, l'UE est ajoutée à sa liste d'UEs et la méthode renvoie true, sinon rien n'est ajouté et false est renvoyé.
- méthodes get et set pour tous les attributs.

5 (3.0 point) : La classe EtDoctorat hérite de la classe Etudiant, elle contiendra les variables et méthodes suivantes :

- directeurThese: chaîne de caractères.
- Un premier constructeur à trois paramètres pour initialiser tous les attributs.
- Un deuxième constructeur qui prend en paramètre le numéro d'étudiant et initialise les autres attributs avec des valeurs par défaut (zéro pour les chiffres et vide pour les chaînes de caractères). Ce constructeur doit appeler le constructeur à trois paramètres.
- une méthode toString qui retourne « Etudiant Numero » numero « Doctorat, directeur de These=» directeurThese.
- une méthode ajouterUE qui prend en paramètre l'UE où l'étudiant veut s'inscrire. Les étudiants de doctorat ont le droit de s'inscrire seulement aux UEs de doctorat. Si l'étudiant a le droit de s'inscrire, l'UE est ajoutée à sa liste d'UEs et la méthode renvoie true, sinon rien n'est ajouté et false est renvoyé.
- méthodes get et set pour tous les attributs.

4 (8.0 point) : Créer la classe TestEtudiant qui contiendra méthodes suivantes :

- `public static UE [] initTabUE(int nbUE)`
Initialiser un tableau d'UEs avec «nbUE» cases. La description de chaque UE sera le mot UE plus un numéro entier aléatoire dans l'intervalle [0, 2000], et le niveau sera choisi au hasard

entre 1-licence, 2-master et 3-doctorat.

- `public static Etudiant [] initTabEtu(int nbEtudiants)`
Initialiser un tableau d'Etudiants avec «nbEtudiants» cases. Le numéro d'étudiant sera un nombre aléatoire entier entre [1, 10000]. Nous allons créer à peu près 20% d'étudiants de doctorat, 30% de master et 50% de licence. Le parcours des étudiants de licence sera choisis aléatoirement dans l'ensemble ["Math", "Physique", "Info", "Chimie", "Biologie"]. La spécialité dans ["E1", "E2", "E3", "E4", "E5", "E6"] et les directeurs de thèses dans ["Le Brand", "Stion", "Ficher", "Bonon"]. Les étudiants de licence peuvent s'inscrire au maximum à 10 UEs, ceux de master à 6 et ceux de doctorat à 3.
- `public static Etudiant [] inscription(Etudiant [] etds, UE [] ues)`
Cette méthode aura comme paramètres un tableau d'Etudiants et un tableau d'UE. Vous devez inscrire chaque étudiant à une UE qui est choisit au hasard. Si l'UE ne correspond pas au niveau de l'étudiant vous devez choisir une autre jusqu'à pouvoir inscrire l'étudiant.
- `public static void afficherInscription(Etudiant [] etds)`
Cette méthode doit afficher les données de chaque étudiant et les UEs auxquelles il est inscrit. Vous devez utiliser les methodes toString et afficheListeUEs d'Etudiant.
- `public static Etudiant [] setNotes(Etudiant [] etds)`
Cette méthode doit attribuer trois notes (aléatoires entre [0, 20] à chaque UE des étudiants.
- `public static void afficherNotes(Etudiant [] etds)`
Cette méthode doit afficher les notes des étudiants de licence ainsi que la moyenne.
- `public static void main (String args [])`
Appeler chaque méthode définie précédemment. La méthode inscription doit être invoquée deux fois.
Créez 30 UEs et 100 étudiants.

Bonne chance ☺