

Nom de l'étudiant :

Prénom de l'étudiant :

Groupe :

Examen de Génie Logiciel

Durée 2H, Documents non autorisés

Partie 1 : Introduction au génie logiciel

1) Citer les quatre valeurs de l'approche agile en développement logiciel :

- 1- adaptation au changement
- 2- collaboration avec client
- 3- les individus et leurs interaction
- 4- des logiciels fonctionnelle

2) Citer quatre principes de l'approche agile en développement logiciel :

- 1- client et les développeurs doivent collaborer
- 2- + ils satisfaire le client avec des livraisons rapide
- 3- la bonne conception et l'extensibilité technique
- 4- simplifier au max

Partie 2 : Processus de développement Agile

augment l'agilité

QCM : Recopiez les bonnes réponses dans le tableau de réponses de la page 10 obligatoirement

1. Quelles sont les activités typiques réalisées lors d'une réunion de revue de Sprint ?

- A) Discussion des possibilités d'amélioration dans l'organisation du travail des développeurs pour les Sprints à venir.
- B) Présentation de la performance du projet aux parties prenantes.
- ☒ C) Démonstration de l'incrément de développement réalisé par l'équipe Scrum aux parties prenantes.
- D) Utiliser la technique de planning Poker pour faire l'estimation de la complexité des tâches
- E) Utiliser la technique de priorisation Moscow pour classer les user stories selon la priorité.

2. Dans Scrum, quand est ce qu'on peut faire les activités de spécification, conception et d'implémentation ?

- A) Chaque user Story
- ☒ B) Chaque Sprint
- C) Chaque Rétrospective
- D) Tous les choix précédents

3. Le pouvoir d'annuler le Sprint appartient uniquement au Product Owner. Il peut demander l'annulation d'un sprint en cours lorsque les éléments du sprint ne sont plus intéressants pour les clients finaux, c'est-à-dire lorsque les éléments de ce sprint sont devenus obsolètes.

- ☒ A) Oui, cette proposition est correcte en Scrum
- B) Non, cette proposition est fausse en Scrum. Ce sont les développeurs et le Scrum master qui ont le droit d'annuler un sprint.

4. Le Product Backlog doit être ordonné sur la base de :

- ☒ A) La valeur des incréments à livrer. La valeur d'un incrément à livrer est déterminée par le Product Owner et c'est lui qui est responsable de l'ordonnement du Product Backlog.
- B) La complexité et la facilité à réaliser des fonctionnalités. On commence par les user stories jugées difficiles et on laisse ceux jugés faciles vers la fin du projet.
- C) Ce sont les clients qui précisent la priorité en contactant directement le Scrum Master et/ou les développeurs sans avoir besoin de passer par le Product Owner.

5. En Scrum, la durée typique du Sprint est de 1 à 4 semaines, laissant place à des événements spécifiques comme la planification du Sprint (8 heures), le Daily Scrum (15 minutes), la revue de Sprint (4h) et la rétrospective de Sprint (3h), chacun adapté en fonction de la durée du Sprint.

- ☒ A) Oui, cette proposition est correcte en Scrum
- B) Non, cette proposition est fausse en Scrum. Un sprint peut aller jusqu'à 8 semaines.
- C) Non, cette proposition est fausse en Scrum. L'événement de sprint review peut durer jusqu'à **cinq** jours pour faire des démonstrations détaillées aux parties prenantes sous forme d'une formation complète sur l'utilisation des incréments développés par l'équipe Scrum.
- D) Non, le Daily Scrum peut durer jusqu'à 4 heures pour aider et débloquer un développeur qui a travaillé sur une fonctionnalité pour laquelle les tests ne sont pas passés avec succès.

6. Scrum a besoin d'un Scrum Master pour favoriser un environnement où le Product Owner ordonne le travail à faire pour résoudre un problème complexe dans le Product Backlog. Le Scrum Team transforme une sélection de ce travail en un Increment de valeur lors d'un Sprint. Le Scrum Team et ses parties prenantes inspectent les résultats et s'adaptent pour le prochain Sprint.

- ☒ A) Oui, cette proposition est correcte
- B) Non, en Scrum il y a 5 rôles : Product Owner, Développeur, Product Manager, testeur logiciel.

7. Qui a la responsabilité de mettre à jour les estimations du travail restant à faire dans un Sprint ?

- A) Scrum Master
- B) Product Owner
- ☒ C) Développeurs

8. Le Scrum Master rend service au Scrum Team de plusieurs façons :

Accompagner les membres de l'équipe en matière d'autogestion et de pluridisciplinarité ; Aider le Scrum Team à se focaliser sur la création d'Increments de grande valeur qui répondent à la Definition of Done ; faire en sorte qu'il n'y ait pas d'obstacles pouvant entraver la progression du Scrum Team ; s'assurer que tous les événements Scrum ont bien lieu et sont efficaces, productifs et respectent bien les temps impartis.

- ☒ A) Oui, cette proposition est correcte
- ☐ B) Non, cette proposition est fausse. Le Scrum Master est aussi responsable de la rédaction des user stories et la priorisation du Backlog de produit.

9. Le Sprint Backlog est composé de l'Objectif de Sprint (le « pourquoi »), de l'ensemble des éléments du Product Backlog choisis pour le Sprint (le « quoi »), ainsi que d'un plan d'action pour la réalisation de l'Increment (le « comment »).

- ☒ A) Oui, cette proposition est correcte
- ☐ B) Non, cette proposition est fausse. Le sprint Backlog est une liste d'exigences non fonctionnelles que doivent respecter les user stories à développer dans un sprint.

10. Sprint Planning est une :

- ☒ A) Réunion où l'équipe Scrum sélectionne les éléments du Product Backlog pour le prochain Sprint, définit l'objectif du Sprint et crée le Sprint Backlog.
- ☐ B) Réunion de négociation de prix de projet avec le client, de précision du budget alloué au projet et de planification mode de paiement des échéances (par exemple : échéance 1 : fin janvier 2025, échéance 2, fin mars 2025, ...)
- ☐ C) Réunion de planification d'une formation sur Scrum où les nouveaux membres de l'équipe Scrum seront formés sur l'approche agile et le Framework Scrum.

11. Si la Definition of Done pour un Increment fait partie des standards de l'organisation, toutes les Scrum Teams doivent la suivre au minimum. Si cela ne fait pas partie des standards de l'organisation, le Scrum Team doit créer sa propre Definition of Done qui soit appropriée pour le produit. Les Developers sont tenus de se conformer à la Definition of Done. Si plusieurs Scrum Teams travaillent ensemble sur un même produit, elles doivent la définir ensemble et s'y conformer.

- ☒ A) Oui, cette proposition est correcte
- ☒ B) Non, cette proposition est fausse. Les développeurs de chaque équipe Scrum créeront et respecteront leurs propres définitions de Terminé (Definition of Done).
- ☐ C) Non, Les développeurs de chaque équipe Scrum créent la sienne et s'assurent que toutes les autres équipes comprennent les différences.

12. Quelle affirmation est vraie concernant le Product Backlog ?

- ☐ A) Il ne doit pas avoir plus de 6 mois de travail en total.
- ☐ B) Il s'agit d'un plan de projet complet à 100% et fixe que l'équipe Scrum s'engage à réaliser et n'autorisera personne à le modifier.

- ☒ C) C'est un artefact vivant qui évolue au fur et à mesure que le produit et l'environnement dans lequel il sera utilisé évoluent. Il peut être modifié selon la demande de nouvelles fonctionnalités par les utilisateurs du produit.

13. Le Planning Poker est une méthode d'estimation collaborative, essentielle dans le cadre d'une équipe Scrum. Il permet d'évaluer l'effort nécessaire pour accomplir les tâches de développement. Comment calcule-t-on généralement la vélocité d'une équipe Scrum ?

- A) En prenant la moyenne des estimations de tous les membres de l'équipe pour chaque user story lors de l'estimation des user stories.
- B) En additionnant les estimations de tous les membres de l'équipe, pour tous les éléments du Product Backlog.
- ☒ C) En calculant la somme M des points d'effort (user story point) des user stories achevées lors des N Sprints précédents dans lesquels cette équipe Scrum a travaillé. La vélocité est obtenue en divisant M par N.
- D) En utilisant une estimation fournie par le Product Owner pour chaque user story.
- E) En divisant le nombre total des user stories achevées par l'équipe au cours du Sprint précédent par le nombre de jours de travail dans ce Sprint.

14. Quel est le but principal du Daily Scrum ?

- A) Discuter en détail de toutes les tâches effectuées depuis la réunion de Sprint Planning.
- ☒ B) Mettre à jour le Scrum Master sur les progrès de chaque membre de l'équipe afin de détecter quels sont les formations complémentaires nécessaires pour les développeurs qui ne terminent pas leurs tâches au délais convenus.
- ☒ C) Identifier les obstacles qui pourraient empêcher l'équipe d'atteindre son objectif du Sprint.
- D) C'est une occasion pour le Product Owner pour remplacer des éléments du Backlog de Sprint par de nouveaux éléments.

15. Lequel des énoncés suivants N'EST PAS l'un des douze principes du Manifeste Agile ?

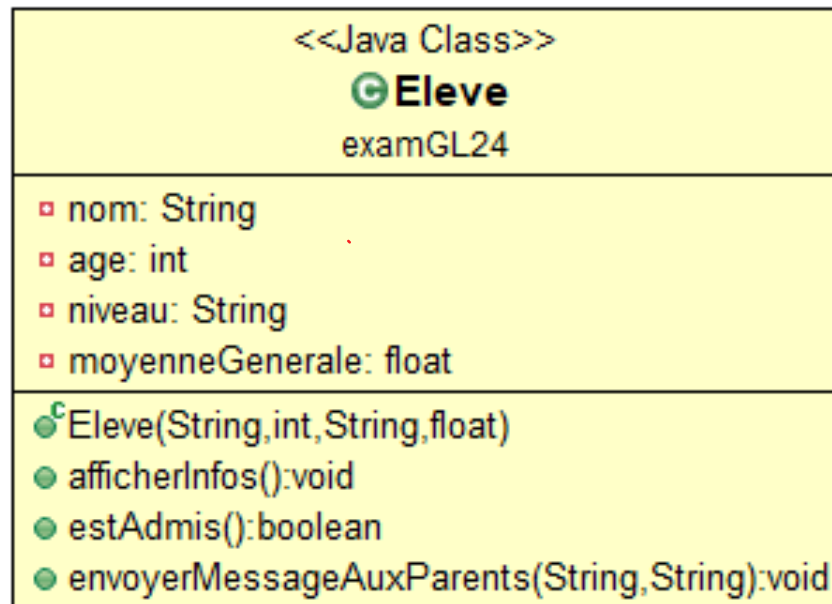
- A) Satisfaire le client est la priorité
- B) Accueillir les demandes de changement
- C) Livrer le plus souvent possible des versions opérationnelles de l'application
- ☒ D) Une documentation complète est la meilleure mesure de progrès

16. L'objectif du Sprint Retrospective consiste à réfléchir à des pistes pour améliorer la qualité et l'efficacité. Le Scrum Team inspecte le déroulement du dernier Sprint en ce qui concerne les individus, les interactions, les processus, les outils et leur Definition of Done. Le Scrum Team identifie les changements les plus utiles pour améliorer son efficacité. Les améliorations ayant le plus d'impact sont abordées dès que possible.

- ☒ A) Oui, cette proposition est correcte
- B) Non, cette proposition est fausse. Le Sprint Restrospective est une réunion de démonstration des incréments réalisés dans un sprint aux parties prenantes.
- C) Non, cette proposition est fausse. Le Sprint Restrospective est une réunion de raffinement du Backlog de produit en détaillant le maximum possible les user stories.

Partie 3 : Qualité et patrons de conception

Exercice 1 : Soit la classe « Eleve » suivante et son code java :

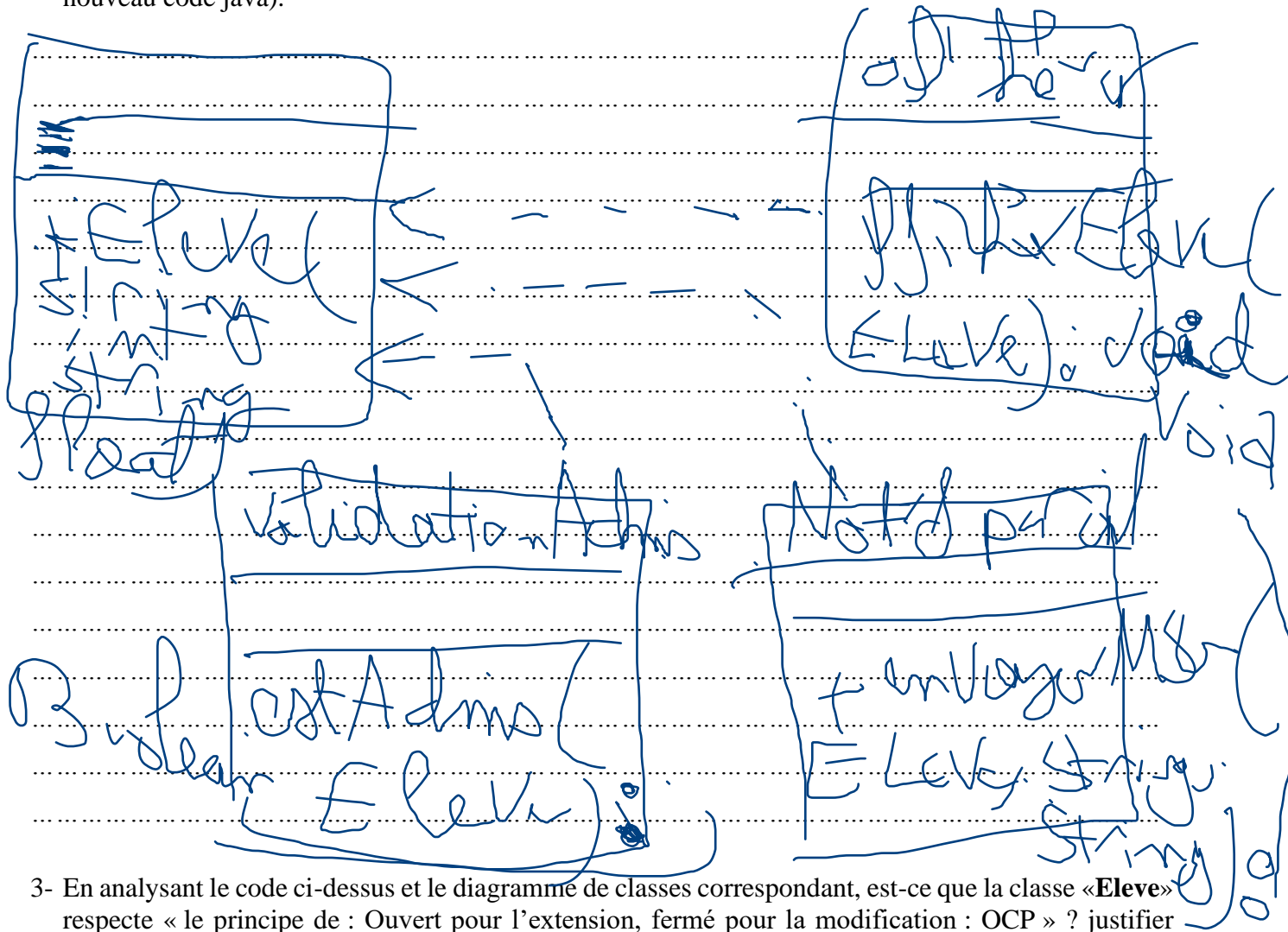


```
1 package examGL24;
2 class Eleve {
3     private String nom;    private int age;
4     private String niveau;    private float moyenneGenerale;
5
6     public Eleve(String nom, int age, String niveau, float moyenneGenerale) {
7         this.nom = nom;    this.age = age;    this.niveau = niveau;
8         this.moyenneGenerale = moyenneGenerale;    }
9
10    public void afficherInfos() {
11        System.out.println("Nom: " + nom); System.out.println("Age: " + age);
12        System.out.println("Niveau: " + niveau);
13        System.out.println("Moyenne générale: " + moyenneGenerale); }
14
15    public boolean estAdmis() {return moyenneGenerale >= 10.0; }
17    public void envoyerMessageAuxParents(String contenu, String methodeEnvoi) {
18        switch (methodeEnvoi) {
19            case "SMS":System.out.println("SMS envoyé aux parents de " +
20                nom + ": " + contenu); break;
21            case "WhatsApp": System.out.println("Message WhatsApp envoyé aux parents de " +
22                nom + ":" + contenu);break;
23            case "Email": System.out.println("Email envoyé aux parents de "+nom+
24                " avec le contenu:" + contenu);break;
25            case "Lettre": System.out.println("Lettre postale envoyée aux parents de " +
26                nom + ": " + contenu); break;
27            default:System.out.println("Autre méthode d'envoi de messages."); break; }} }
```

- 1- En analysant le code ci-dessus et le diagramme de classes correspondant, est-ce que la classe «**Eleve**» respecte « le principe de responsabilité unique : SRP » ? justifier votre réponse ?

non ne respect pas le principe de SRP car cette classe a plusieurs responsabilité et selon le principe de SRP c que une classe devrait avoir qu'une seule rison de changer

- 2- Afin de respecter le principe SRP, diviser la classe d'origine «**Eleve**» en des classes homogènes et cohérentes et donner le **nouveau diagramme de classes** qui respecte le principe SRP (sans faire le nouveau code java).



- 3- En analysant le code ci-dessus et le diagramme de classes correspondant, est-ce que la classe «**Eleve**» respecte « le principe de : Ouvert pour l'extension, fermé pour la modification : OCP » ? justifier votre réponse ?

non la classe eleve ne respect pas le principe de OCP car la methode envoyerMessageAuxParent() utilise un switch/case et pour ajouter une NV communication il faut modifier la classe et selon OCP il faut ajouter des NV communication sans modifier la classe 'ouverte a l'extension fermé a la modif '

- 4- Si la classe «**Eleve**» ne respecte pas « le principe de : Ouvert pour l'extension, fermé pour la modification : OCP », avec quel Design Pattern on peut régler ce problème (cochez la bonne réponse) :

- ☐ Design Pattern Strategy
☒ Design Pattern Decorator

Exercice 2 :

Vous êtes chargé de développer un système de gestion des inscriptions d'élèves dans une école privée. Pour cela, vous devez utiliser le patron de conception Décorateur afin de permettre aux élèves et leurs parents de choisir différentes fonctionnalités supplémentaires lors de leur inscription à l'école.

Pour **simplifier l'énoncé** de cet exercice et se focaliser uniquement sur l'implémentation du Design pattern Decorator, **on fixe le prix de base** de l'inscription à l'école à 2000 MAD et on fixe le prix des fonctionnalités supplémentaires (les décorateurs) actuellement disponibles comme suit :

- ✓ Transport scolaire : Ajoute 500 MAD aux frais d'inscription.
- ✓ Cantine scolaire : Ajoute 350 MAD aux frais d'inscription.
- ✓ Langue supplémentaire : Ajoute 450 MAD aux frais d'inscription.

Votre tâche consiste à implémenter ce système en utilisant le pattern Décorateur et en offrant aux élèves et leurs parents la possibilité de choisir les options qu'ils veulent au moment d'inscription.

On aura besoin d'une interface **Inscription** avec les deux méthodes suivantes :

- **float calculerPrixTotalInscription()** : qui renvoie le prix total à payer pour l'inscription selon les options choisies, comme le montre l'exemple d'exécution ci-dessous.
- **String description()** : qui affiche les détails de l'inscription comme le montre l'exemple d'exécution ci-dessous.

On aura besoin aussi d'une classe **InscriptionScolaire** qui représente l'inscription de l'élève aux cours officiels du ministère.

```
interface Inscription { float calculerPrixtotalInscription(); String description();}

//Classe de base InscriptionScolaire

class InscriptionScolaire implements Inscription {

@Override

public float calculerPrixTotalInscription() { return 2000; }

@Override

public String description() {return " Inscription aux cours officiels du ministère"; }}

////////////////////////////////////
//

//ici sont ajoutées les autres classes.....

public class MainExamenGL {

public static void main(String[] args) {

// Création d'une inscription scolaire de base avec un prix initial de 2000 MAD

Inscription inscription = new InscriptionScolaire();
```



```
// Décoration de l'inscription avec différentes fonctionnalités

inscription = new TransportScolaireDecorator(inscription);

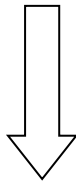
inscription = new CantineScolaireDecorator(inscription);

inscription = new LangueSupplementaireDecorator(inscription);

float prixTotal = inscription.calculerPrixTotalInscription();

System.out.println("Prix total de l'inscription : " + prixTotal + " MAD");

System.out.println("Description de l'inscription:\n" + inscription.description()); }}
```






```
Prix total de l'inscription :3300.0 MAD
Description de l'inscription :
  Inscription aux cours officiels du ministère
  Avec Transport scolaire
  Avec Cantine scolaire
  Avec Langue supplémentaire
```

Travail à faire :

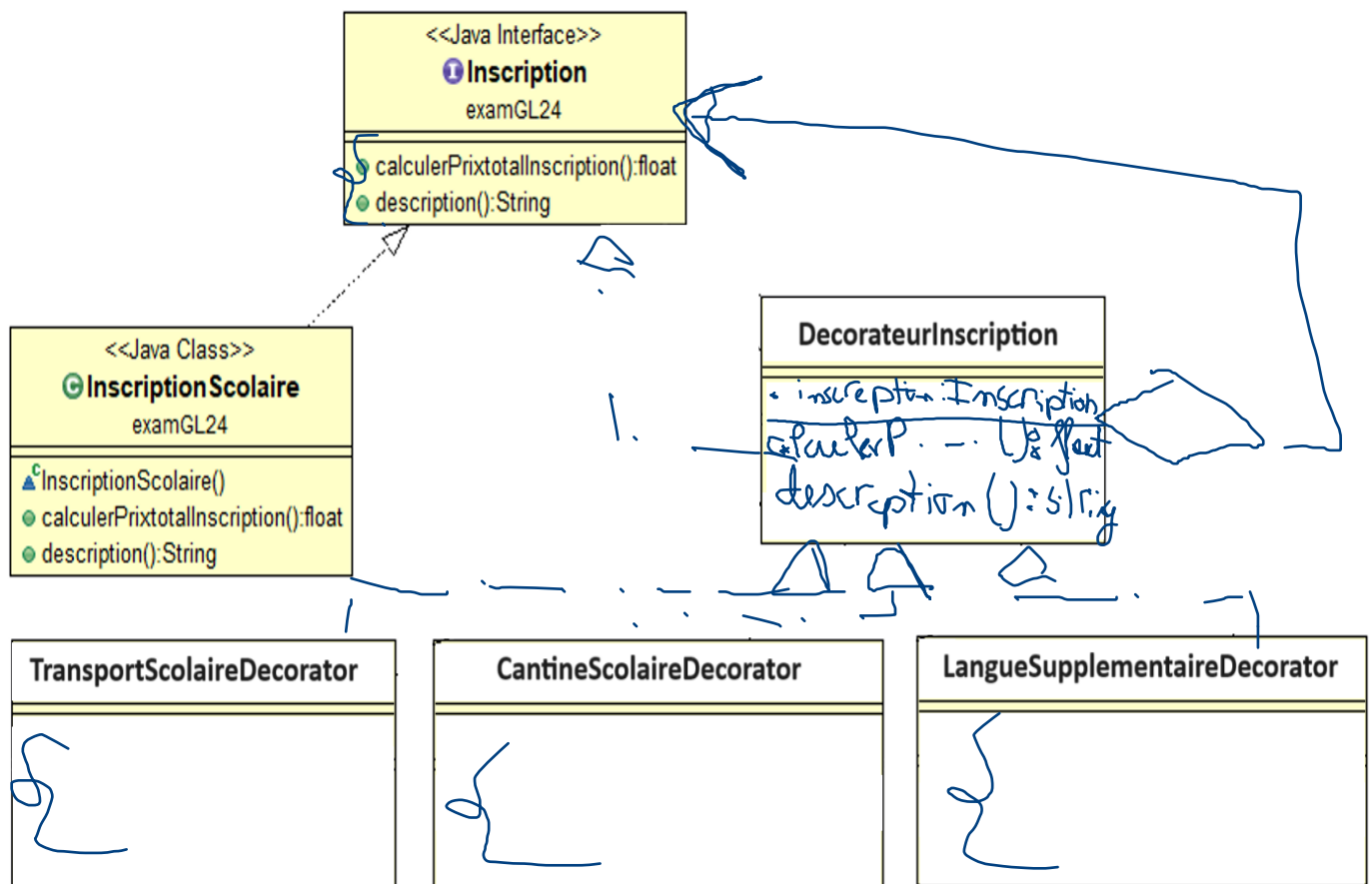
- 1) Compléter le diagramme de classes suivant en précisant les **relations** (héritage, composition, implémentation, ...) entre les différentes classes, **les attributs** et **les méthodes** de chaque classe.

Rappels sur les symboles à utiliser :

Rappels sur les symboles à utiliser :

	Composition
	Héritage/spécialisation (extends)
	Héritage d'interface(implements)

NB : L'interface Inscription peut être remplacée par une classe abstraite



2) Compléter le tableau suivant en précisant le contenu de la classe **DecorateurInscription**

Nom de la classe	Liste des attributs	Liste des méthodes
DecorateurInscription	<pre> inscription: Inscription </pre>	<pre> { } </pre>

Tableau des réponses du QCM :

Cocher la case correspondante en mettant un X comme suit : 

Numéro de la Question	Réponse A	Réponse B	Réponse C	Réponse D	Réponse E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					