

Ecole Supérieure Privée Technologies & Ingénierie

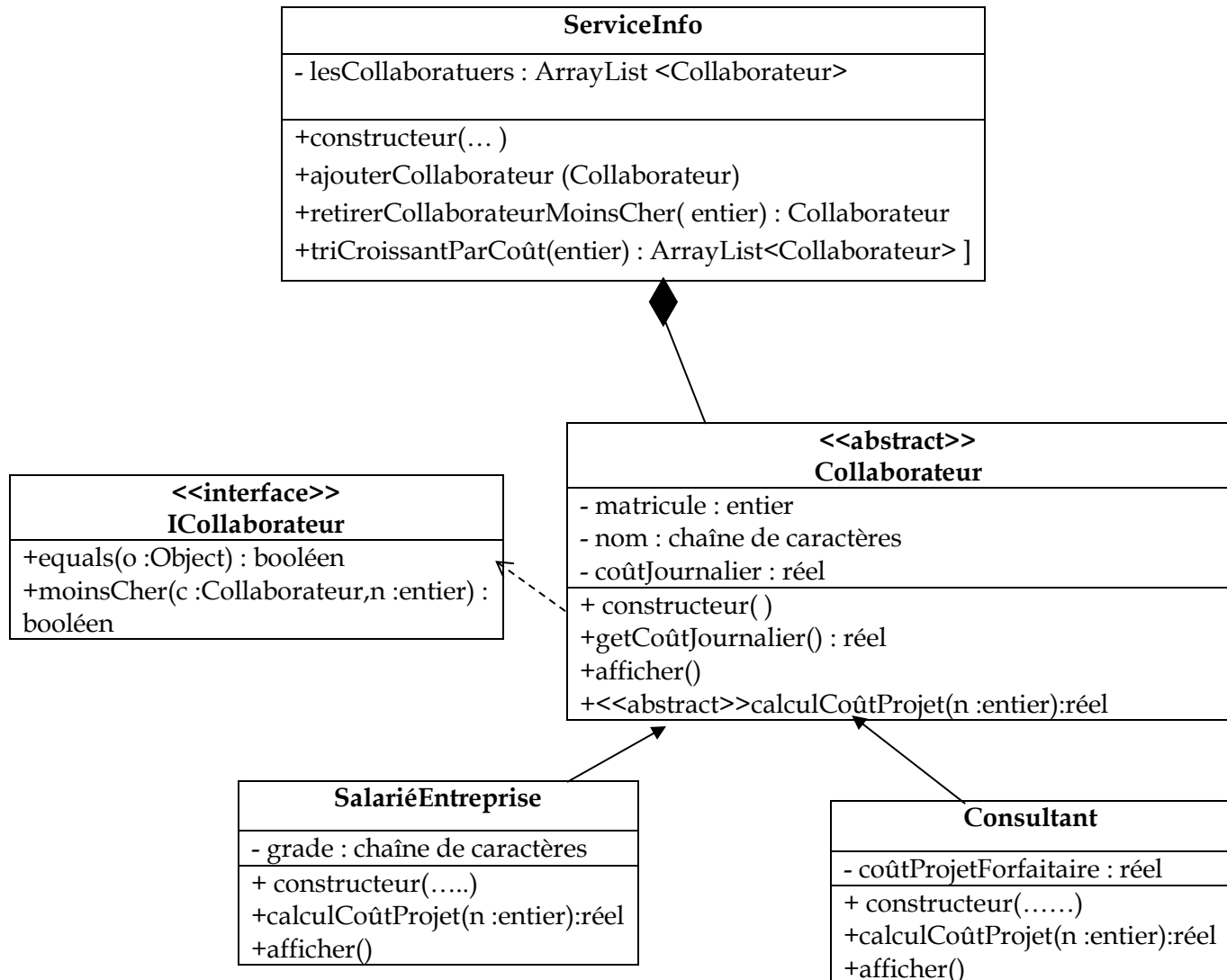
Type d'épreuve : ☐ Devoir ☒ Examen
Enseignants : Mme Cheikh.E, Mr Hassan. K, MrFarhat.M, MrBouhari.S
Matière : Programmation Orientée Objet et Java
Année Universitaire : 2019-2020 **Semestre** : 2
Classe : TIC-1 B/C/D/E/F/G/H/I
Documents : ☒ Non autorisés ☐ Autorisés
Date : 29/06/2020 **Durée** : 1h30mn
Nombre de pages : 4

Exercice1 : (5 points)

<pre> 1 class Exercice1 { 2 public static void main(String[] args) 3 { 4 A x = new A(); 5 B y = new B(); 6 C z = new C(); 7 y.b = 2; 8 z.c = 3; 9 } 10 abstract class A { 11 int a;} 12 class B extends A { 13 int b; 14 public B(int b){ 15 this.b = b;} 16 } 17 class C extends A { 18 final double c = 1; 19 } 20 abstract class D extends A { 21 double d; 22 int operation(int a) { 23 return (a * 2); 24 } 25 abstract int calcul(int b) { } 26 abstract void afficher(); } </pre>	<p>1.1 Question (2.5 Points)</p> <p>Le programme suivant comporte des erreurs. Trouver les lignes de ces erreurs et les expliquer.</p>
<p>Soit la classe Date suivante :</p> <pre> class Date { int jour; int mois; int annee; Date(int j,int m, int a) { jour = j; mois = m; annee = a; }} </pre>	<p>1.2 Question (2.5 Points)</p> <p>Modifiez le constructeur de cette classe de manière à ce qu'il lève et propage (vous utilisez throw et throws) d'une nouvelle exception nommée DateException dans le cas où le jour ou le mois ou l'année ne correspondent pas à une date valide (on supposera que les mois ont tous 30 jours)</p>

Problème : (15 Points)

Une entreprise dispose d'un service informatique, qui développe des projets pour toute l'entreprise. Ce service informatique a besoin de connaître précisément le coût de chacun des projets qu'elle réalise. Soit le diagramme de classes suivant :



Avec : L'interface **ICollaborateur** est l'interface que doit implémenter la classe **Collaborateur** pour vérifier si deux Collaborateurs sont identiques cad ont le même matricule et aussi pour vérifier si le coût de projet qui dure n jours pour un Collaborateur donné est inférieur ou égal au coût de projet de n jours pour un autre Collaborateur.

Partie n°1 : (12 Points)

- Écrire le code java de l'interface **ICollaborateur** (0.5 Point)
- Écrire le code java de la classe **Collaborateur** sachant que : (3 points)
 - **constructeur(...)** : initialise tous les attributs de la classe par des valeurs passées en paramètres.
 - **equals(.....)** : vérifie si deux collaborateurs ont le même matricule
 - **moinsCher(.....)** : vérifie si un collaborateur donné a un coût de projet inférieur ou égal au coût du projet d'un autre collaborateur et ceci pour le même nombre de jours n.
 - **afficher()** : affiche tous les attributs d'un Collaborateur séparés par un espace.
- Écrire **uniquement** le code de la classe **Consultant**. (la classe SalariéEntreprise on suppose qu'elle est déjà définie). Sachant que :
 - **constructeur(...)** : initialise tous les attributs de la classe par des valeurs passées en paramètres.
 - **afficher()** : affiche toutes les informations concernant un Consultant avec en plus le coût d'un projet qui durera 10 jours par exemple.
 - **calculCoûtProjet (n : entier)** : calcule le coût d'un projet qui a duré n jours pour un consultant comme suit : Le coût de projet de n jours égal n fois le coût journalier plus le coût forfaitaire indépendamment du nombre de jours n. (cad $n \times \text{coûtJournalier} + \text{coût forfaitaire}$)(1.5 Points)
- Ecrire le code de la classe **ServiceInfo** caractérisée par lesCollaborateurs qui est un ArrayList qui mémorise tous les collaborateurs du service.
Les méthodes à définir sont :
 - **constructeur ()** : crée l'attribut lesCollaborateurs. (0.5 Point)
 - **ajouterCollaborateur(...)** : ajoute un collaborateur donné à lesCollaborateurs et ceci dans le cas où ce dernier n'existe pas déjà dans l'ArrayList. (1.5 Point)
 - **retirerCollaborateurMoinsCher(...)** : retire et retourne le collaborateur le moins cher de l'ArrayList (cad celui qui a le coût de projet pour n jours le moins cher). Dans le cas d'égalité entre deux collaborateurs elle retire le premier trouvé le moins cher. (3 Points)
 - **triCroissantParCoût ()** : renvoie un ArrayList contenant tous les collaborateurs, triés par ordre croissant de coût pour un projet de n jours. (2 Points)

Partie n°2 : (3 Points)

. Écrire le code de la classe Test réduite à une méthode main qui permet de :

- Créer un service informatique et ajouter les deux collaborateurs suivants :
 - c1 (2, "Ben salah", 60, " concepteur ")
 - c2 (1, "Ben mrad", 50, 100)**(1.5 Point)**
 - Trier Les collaborateurs par ordre croissant de leurs coûts de projet pour une période de 10 jours. **(0.75 Point)**
 - Afficher alors tous ces collaborateurs avec cet ordre. **(0.75 Point)**
-

Annexe ArrayList

- Déclaration d'une liste d'objets de type Type : ArrayList <Type>Lst;
- Constructeur créant une liste vide : Lst=new ArrayList<Type> ();
- Ajout d'un élément **t** à la fin de la liste : Lst.add(t), la méthode retourne un booléen
- Insertion d'un élément **t** à la position **i** : Lst.add(i-1, t)
- Retourner élément à l'indice **i** : Lst.get(i)
- Placer un élément **t** à l'indice **i** : Lst.set(i,t)
- Nombre d'éléments de la liste : Lst.size()
- Supprimer un élément **t** de la liste : Lst.remove(t), la méthode retourne un booléen
- Tester si un élément existe dans la liste : Lst.contains(t), la méthode retourne un booléen

Bon travail