|  |  |  |  |
| --- | --- | --- | --- |
|  | **Année Universitaire : 2014-2015**  **Devoir surveillé** | | |
| **Module** : Conception orientée objet et Programmation Java | | | **Documents autorisés** : Non |
| **Enseignants** : Equipe JAVA | | | **Nombre de pages** : 5 |
| **Date** : 01/11/2014 | | **Heure :** 9h | **Durée** : 1h |
| **Classes** : **3** info A **,4** Infini, **3** SIGMA | | |  |
| **Nom & Prénom :** | | |  |

**NB : L’étudiant doit répondre sur cette feuille et la rendre**

**Exercice I : QUESTIONS THERORIQUES *(4.5 pts)***

**Question 1 :** Identifier dans la classe **Candidat** pour la méthode **voter ()** le cas de la surcharge ou redéfinition ( 0.5 pt )

public Candidat **extends** Citoyen {

private boolean inscrit;

(1) public void voter () {

inscrit = true;

}

(2) public void voter (int identifiant) {

…

} }

}

public class Citoyen {

…

**public void voter()** {

…

}

}

**Réponses :**

(1) redéfinition

(2) surcharge

**Question 2**: Corriger le code suivant: (1 pt)

public class Test {

public class Calculatrice {

public int calcul (int x) {

return (x \* 2);

}

}

pubc static void Main(String[ ] args) {

int lix = args[0] ;

Calculatrice.calcul(x);

}

}

**choisissez la première ou la deuxième solution**

public class Calculatrice {

public static int calcul (int x) {

return (x \* 2);

}

}

public class Test {

pubc static void Main(String[ ] args) {

int lix = args[0] ;

Calculatrice calc = new Calculatrice() ;

calc.calcul(x);

}

**NB : L’étudiant doit répondre sur cette feuille et la rendre**

**Question 3**: Encapsulation des attributs/méthodes : Complétez le tableau suivant en mettant vrai ou faux dans chaque cas (1 pt)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***public*** | ***Private*** | ***protected*** | ***default*** |
| ***La classe elle même*** | ***Vrai*** | ***Vrai*** | ***Vrai*** | ***Vrai*** |
| ***Une classe du même package*** | ***Vrai*** | ***Faux*** | ***Vrai*** | ***Vrai*** |
| ***Une sous classe du même package*** | ***Vrai*** | ***Faux*** | ***Vrai*** | ***Faux*** |
| ***Une classe d’un autre package*** | ***Vrai*** | ***Faux*** | ***Faux*** | ***Faux*** |
| ***Une sous classe d’un autre package*** | ***Vrai*** | ***Faux*** | ***vrai*** | ***Faux*** |

**Question 4**: Définir brièvement les termes suivants :(1 pt)

JDK kit de développement java qui contient la JRE et la JVM

JRE La JRE, c'est le Java Runtime Environment, ça permet à tous programme Java de s'exécuter sur la machine.

JVM : la Java Virtual Machine, donc la machine virtuelle de Java. Elle est contenue dans la JRE

**Question 5**: Cochez la bonne réponse : (1 pt)

public class From {

public From (String s) {

System.out.print("F");

}}

public class Cercle extends From {

public Cercle (String s) {

System.out.print("C"); }

public static void main(String [] args) {

new Cercle ("D");

System.out.println("s");

}

}

**Quel est le résultat?**

A. FC

B. CF

C. FCD

D. CFD

**E. Erreur**

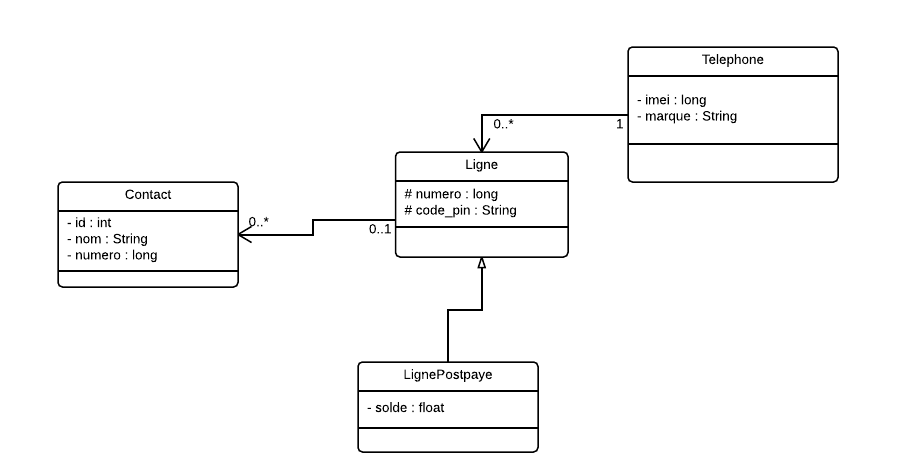
**Expliquer votre choix :**

La première instruction à exécuter dans le constructeur de la classe Cercle est super(), alors que dans la classe Form le constructeur par défaut n’existe pas.

**Exercice II : PROBLEME *(15,5pts)***

L’instance nationale des télécommunications souhaite informatiser le système de gestion des opérateurs du réseau mobile en Tunisie dans le but d’améliorer son système d’information. Nous proposons de développer un programme pour gérer les opérateurs mobiles agréés à l’instance nationale des télécommunications, ainsi que la gestion de lignes téléphoniques et leurs abonnées (contacts).

Vous trouverez ci-dessous une partie du diagramme de classe modélisant ce processus.



**En annexe vous trouvez l’implémentation de la classe principale (Main.java) afin de vous guider pour la création des différentes classes et méthodes (constructeurs**, **type de retour des méthodes) ainsi le résultat d’affichage;**

**Question 1 :**

Créer la classe **Contact**  (ses attributs et constructeurs)ainsi que les méthodes suivantes**(3pts)**

**RQ : Les getters et setters seront implémentés seulement dans la classe Contact pour l’attribut « id », pour les autres attributs on suppose qu’ils existent.**

1. **La méthode comparerContact(Contact c) :** deux contacts sont identiques dans le cas où ils ont le même identifiant (id) et le même nom aussi
2. **La méthode toString() :** permet de retourner les attributs d’un contact

**Question 2 :** Créer la classe **Ligne (**ses attributs, constructeurs et méthodes nécessaires), ainsi que les méthodes suivantes, sachant que cette classe ne doit pas être instanciable. **(1 pt)**

1. **ajouterContact(Contact c) :** en prenant en considération que  **(2 pts)**
   1. le *nombre maximale* des contacts dans une **ligne** est 200 ;
   2. un contact ne peut être affecté à une ligne donnée qu’une seule fois

*(Exemple : pour la ligne ligne1 on peut ajouter qu’une seule fois le contact c1)*

1. **toString() :** retourne une ligne ainsi que tous les contacts qui lui est associés **(1.5pts)**
2. **calculerSolde(int duree):** le traitement sera précisé dans la question suivante. **(0.5 pts)**

**Question 3 : (3pts)**

Implémenter la classe **LignePostPaye (**ses attributs et constructeurs) et les méthodes :

1. **calculerSolde(int duree) :**

Le solde est calculé comme suit :

**solde – (cout \* duree)** sachant que lecout de la minute est 0.060DT

1. **toString() :** retourne les détails d’une ligne post payéavec le solde

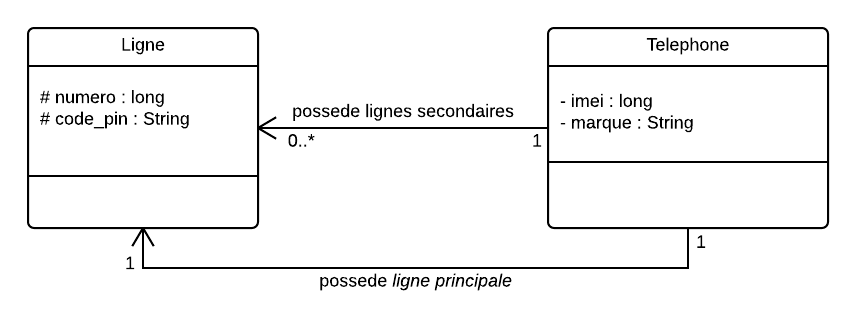
**Question 4: (2pts)**

Implémenter la classe **Telephone** (ses attributs et constructeurs nécessaires), sachant que le nombre maximale de lignes doit être mentionné à la création du téléphone **(voir annexe)**

Ajouter la méthode suivante: **chercherContact(long numreo)**

**Question 5 : (2.5pts)**

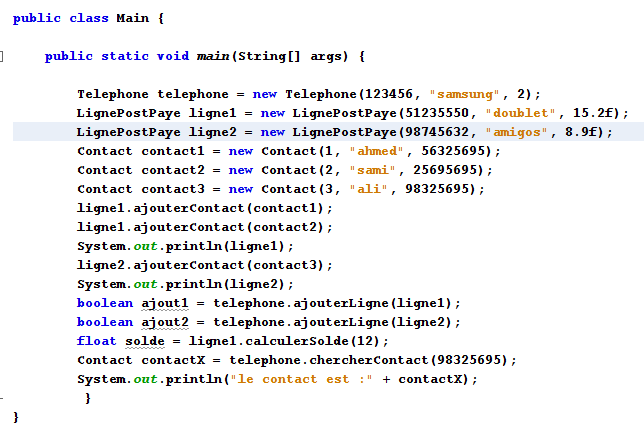
Etant donné qu’un téléphone peut avoir qu’une seule ligne principale (active), implémenter la relation désignée par le rôle « *possède ligne principale* » dans la classe **Telephone** **(voir figure)**



Tester dans la méthode main la création d’un téléphone qui possède une seule ligne principale et qui peut supporter 2 lignes secondaires.

***Bon Travail***

***Annexe : la classe Main.java***

******

***Affichage Final :***

