

#### **EXERCICE 4 : RIP**

4 Pt.

- 1) Quelles actions résultent de la saisie de la commande **network 192.168.1.0** en mode de configuration RIP sur un routeur ? (choisir la bonne réponse)

  - a) La table de routage est créée dans la mémoire vive du routeur.
  - b) L'adresse réseau 192.168.1.0 est annoncée aux routeurs voisins.
  - c) Le processus RIP est interrompu et toutes les configurations RIP existantes sont effacées
  - d) Une demande de mises à jour de routage est envoyée aux routeurs voisins.

2) Donner la signification de chaque champ d'une ligne dans la table de routage

**A** **B** **C** **D** **E** **F**  
R 172.16.0.0/25 [120/1] via 192.168.1.1 00:00:12 Serial0/1/0

<u>champ</u>	<u>Signification</u>
A	..... ..... .....
B	..... .....
C	..... .....
D	..... .....
E	..... .....
F	..... .....

EXAMEN

Matière: Protocoles et services Réseaux Calculatrice Scientifique: Autorisée  
Enseignants: Naïma Daghfous. Durée: 1h 30'  
Classes :SEM21

## Documents: Non autorisés

Durée: 1h 30

Date: JANVIER/2021

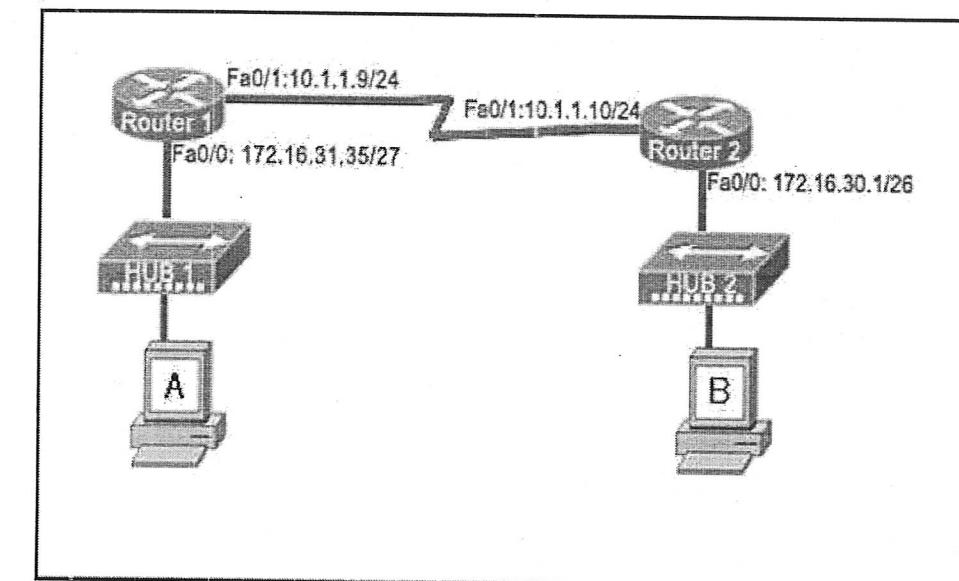
*Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.*

Prénom \_\_\_\_\_ NOM \_\_\_\_\_  
Classe \_\_\_\_\_

## **EXERCICE 1 : ROUTAGE STATIQUE**

**4.5 Pts**

Soit l'illustration suivante



Donner une adresse et un masque pour le Poste A :.....

Donner une passerelle par défaut pour le Poste A :

Donner une adresse et un masque pour le Poste B :

Donner une passerelle par défaut pour le Poste B :

..... Configurer des routes statiques ..... Rute 1 et Rute 2 .....

.....

NOM \_\_\_\_\_  
PRENOM \_\_\_\_\_  
CLASSE \_\_\_\_\_

**EXERCICE 2 : ADRESSE IPv6**
**4 Pts**

Donner le type de l'adresse IPv6

l'adresse IPv6	type
::1	
FC00:22:A:2::CD4:23E4:76FA	
FE80::3201:CC01:65B1	
2001:0DB8:1:ACAD::FE55:6789:B210	
FF02::2	
2033:DB8:1:1:22:G33D:259A:21FE	
FD00::DB7:4322:A231:67C	
FF02::1:FFAB:CDEF	

**EXERCICE 3 : DHCP**
**7.5 Pts**

Un analyseur de messages échangés sur un réseau local Ethernet/IP donne le résultat suivant. Il est constitué d'une suite de lignes correspondant à un message observé sur le réseau local. On trouve un numéro d'ordre du message observé, la date de l'observation en seconde, les adresses IP source et destination, le nom du protocole pour lequel le message a circulé et le type du message

No	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover
2	0.001182	192.168.0.247	192.168.0.5	ICMP	Echo request
3	0.342454	192.168.0.247	192.168.0.5	DHCP	DHCP Offer
4	0.344405	0.0.0.0	255.255.255.255	DHCP	DHCP Request
5	0.348264	192.168.0.247	192.168.0.5	DHCP	DHCP ACK
6	0.353014	CIS_a8:52:24	Broadcast	ARP	Who has 192.168.0.5?
Tell 192.168.0.5					

1) Pour le message numéro 1 de la trace expliquez la signification des adresses IP source et destination (pourquoi selon vous utilise t'on ces adresses dans cet échange) ?

Adresse IP source :.....

Adresse IP destination :.....

2) Pour le message numéro 3 de la trace expliquez la signification des adresses source et destination (à quoi correspondent ces adresses) ?

Adresse IP source :.....

Adresse IP destination :.....

Comment est-il possible que ce message (numéro 3 ) parvienne correctement à son destinataire ?

3) Pour le message numéro 4, expliquez la signification des adresses IP source et destination (pourquoi avoir choisi les adresses IP qui apparaissent dans le message 4)?

Adresse IP source :.....

Adresse IP destination :.....

4) Dans cette trace, le message numéro 2 semble ne pas être relié à une attribution d'adresse DHCP. Cependant ce message a circulé aussi dans le cadre de l'attribution d'adresse. Pourquoi le message numéro 2 a-t-il circulé (quel est le but poursuivi dans la circulation de ce message) ?

5) De la même façon, décrivez pourquoi le message numéro 6 a circulé dans cette trace?

**(Rappel)**

- le protocole ARP permet de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP
- Le protocole ICMP (Internet Control Message Protocol) est un protocole utilisé pour véhiculer des messages de contrôle et d'erreur. Exemple de messages ICMP le PING

# EXAMEN

Matière: POO

Date : Janvier 2021

Enseignante: Imene Sghaier

Durée: 1h30'

Classe : SEM21

Documents & Calculatrice: Non autorisés

NB : Le sujet contient 2 pages. Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.

Écrivez une classe nommée **Phone** pour représenter un téléphone. Cette classe sera une classe parente pour plusieurs types de téléphones. La classe Phone est caractérisée par un numéro de téléphone (chaîne de caractères), un fournisseur de télécommunications (chaîne de caractères), un crédit de recharge d'argent (en dinars) pour payer les appels et un journal d'appels contenant les 100 derniers numéros appelés (tableau de chaînes de caractères).

Sachez que tous les objets de la classe Phone possèdent des numéros gratuits en communs quelque soient leurs fournisseurs de télécommunication. Ces numéros sont les suivants :

193	197	198
-----	-----	-----

La classe Phone doit également définir les méthodes suivantes:

- ✓ Un **constructeur** qui accepte le numéro de téléphone et le nom du fournisseur de télécommunications.
- ✓ Des accesseurs en lecture pour le numéro et le fournisseur de télécommunications.
- ✓ Une méthode **toString** qui renvoie une chaîne
- ✓ Une méthode **loadCredit** qui accepte un montant de recharge et l'ajoute au crédit de charge d'argent du téléphone.
- ✓ Une méthode **call** qui accepte un numéro de téléphone et l'ajoute au journal des appels effectués. Cette méthode doit :
  - Vérifier tout d'abord s'il s'agit ou non d'un numéro gratuit, dans ce cas le prix de cet appel ne sera pas crédité du solde du téléphone mais l'appel sera enregistré dans le journal
  - Si le numéro n'est pas gratuit la méthode call doit vérifier s'il reste encore du crédit pour effectuer l'appel : on va supposer que tous les appels coûtent 0.1 dinars quelque soit la durée et le fournisseur de l'appel. Si le crédit n'est pas suffisant une Exception de type **CreditException** est levée.
  - vérifier s'il reste encore de la place vide dans le tableau des appels effectués sinon la méthode call doit écraser l'appel situé à la première case du tableau (le plus ancien appel) en faisant un décalage gauche de toutes les cases du tableau.

```

class CreditException /*A compléter*/ {
    CreditException(){
        //A compléter
    }
    CreditException(String s){
        //A compléter
    }
}

```

1. Ecrivez la classe **Phone**
2. Complétez le code de la classe **CreditException**
3. Écrivez la classe **SmartPhone** en tant que sous-classe de la classe Phone.

Un smartphone est caractérisé par :

- Une longitude et une latitude de type double
- Une adresse IP de type **adresseIP**

```

public class AdresseIP {
    private int[] octet;

    public AdresseIP (int o1,int o2, int o3, int o4) {
        octet = new int[] {o1,o2,o3,o4};
    }

    public String toString(){
        return octet[0] + "." + octet[1] + "." + octet[2] + "."+ octet[3];
    }
}

```

- Un constructeur qui accepte le numéro de téléphone, le nom du fournisseur et l'adresse IP.
- Une méthode **toString** qui renvoie une chaîne
- La classe **Smartphone** implémente l'interface **GPS**

```

interface GPS{
    void getCoordinates();
    //Méthode permettant d'afficher les coordonnées géographiques d'un dispositif
}

```

4. Sachant qu'une **AdresseIP** est composée de 4 nombres entiers (4 octets) entre 0 et 255, séparés par des points et notées sous la forme **xxx.xxx.xxx.xxx** Par exemple, 194.168.1.7 est une adresse IP donnée. Modifiez le code du constructeur de la classe **AdresseIP** en prenant en compte les cas d'erreurs pour les 4 octets. En cas d'erreur une exception de type **ExceptionIP** est levée.

```

public AdresseIP(int o1,int o2, int o3, int o4) // à compléter //
{
    if (o1 < 0 || o1 > 255)
        { /* à compléter */ }
    /* A compléter suivant le même modèle */
    octet = new int[] {o1,o2,o3,o4};
}

```

5. Donnez un exemple de code de la classe **ExceptionIP**
6. Ecrivez la classe **Test** muni d'une méthode main dans laquelle vous allez :
  - a. Créer un objet **adr** de type **AdresseIP**
  - b. Créer un objet de type **Smartphone** caractérisé par **adr** comme **AdresseIP**, « 98211674 » comme numéro et « TT » comme fournisseur
  - c. Afficher les caractéristiques de l'objet **Smartphone** créé
  - d. Appeler le premier numéro gratuit sur l'objet **Smartphone** créé
  - e. Ajouter 10 dinars au solde du **smartphone** créé
  - f. Appeler le numéro 72 220 051

**NB :** Vous devriez prendre en compte les Exceptions interceptées.

*Bon travail*

- 5) Afficher la meilleure note de stage PFE par département.
- 6) Afficher les détails des stages non valides (code stage, nom et prénom de l'étudiant, date de la soutenance, note, membres du jury)
- 7) Afficher les groupes dont la moyenne des notes des stages perfectionnement est supérieures à 12,00 durant l'année universitaire 2019/2020 (des stages de type perfectionnement qui débutent le 1<sup>er</sup> février 2019).
- 8) Afficher la liste des étudiants (nom, prénom, groupe) qui n'ont effectué aucun stage.

## **EXERCICE 2 : NORMALISATION**

**(6 POINTS)**

Le propriétaire d'une auto-école désire informatiser le fonctionnement de son travail. Il a collecté toutes les informations nécessaires dans la relation R suivante :

R (idCli, numSeance, codeExam, nomCli, adrCli, dateNaiss, idIns, nomIns, gradeIns, dateSeance, heure, duree, questions, reponses, nbErreurs, dateExam, resultat)

Sachant que :

- Un client est identifié par un numéro (idCl) et il est caractérisé par son nom (nomCli), son adresse (adrCli), sa date de naissance (dateNaiss) et l'identifiant de son instructeur (idIns).
- Un instructeur identifié par idIns est décrit par son nom (nomIns) et son grade(gradeIns).
- Une séance d'apprentissage de code, identifiée par son numéro (numSeance), est caractérisée par sa date (dateSeance), son heure de début (heure), sa durée (duree), la liste des questions révisées (questions) et leurs réponses (reponses).
- Un client suit plusieurs séances d'apprentissage. Pour chaque séance, on enregistre le nombre d'erreurs (nberreurs) faits par ce client.
- Un examen de code de la route est identifié par un code (codeExam) et il est caractérisé par la date de son passage (dateExam).
- Un client peut passer plusieurs examens de code. Pour chaque fois, on enregistre le resultat final (resultat).

Travail demandé :

- 1) Normalisez la relation R à la 3<sup>ème</sup> forme normale 3NF en expliquant toutes les étapes.
- 2) Indiquez le schéma relationnel final obtenu.



## EXAMEN

Matière : BASE DE DONNEES Documents : Non autorisés  
 Enseignants : H. Haddaji, K. Bouarada, A. Zaich, S. Guerbouj, S. Majdoub, M. Khemiri Durée: 1h30  
 Classes : DS12, RSI2, SEM2, MDW2 Date : Janvier 2021

*Le Devoir comporte 3 pages  
 Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.*

## EXERCICE 1: LANGAGE SQL

(14 POINTS)

Soit le schéma relationnel suivant :

Etudiant (n\_etud, nom, prenom, adresse, mail, tel, groupe, departement)

Stage (code\_stage, type\_stage, entreprise, date\_deb, #etudiant, #encadrant)

Soutenance (code\_sout, #code\_stage, #code\_etud, date\_soutenance, #encadrant, #rapporteur, #president, decision, note)

Enseignant (n\_ens, nom, prenom, grade, mail, tel, adresse, departement)

Sachant que :

Attribut	Type de donnée	Rôle
<u>n_etud</u>	Entier	Le numéro de l'étudiant
Nom	Chaine de caractères	le nom de l'étudiant
Prenom	Chaine de caractères	Le prénom de l'étudiant
Adresse	Chaine de caractères	L'adresse de l'étudiant
Mail	Chaine de caractères	Le mail de l'étudiant
Tel	Chaine de caractères	Le numéro de téléphone de l'étudiant
Groupe	Chaine de caractères	Le groupe de l'étudiant (DS21, TI14 ...etc)
departement	Chaine de caractères	Le département de l'étudiant (Mécanique, TI, ...)
<u>code_stage</u>	Entier	Le code du stage
type_stage	Chaine de caractères	Le type de stage (initiation, pfe, ...)
Entreprise	Chaine de caractères	L'entreprise accueillant le stage
date_deb	Date	La date de début du stage
Etudiant	Entier	Le numéro de l'étudiant qui fait le stage
Encadrant	Entier	Le numéro de l'enseignant qui encadre le stage

<u>code_sout</u>	Entier	Le code de la soutenance
<u>code_stage</u>	Entier	Le code du stage
<u>code_etud</u>	Entier	Le numéro de l'étudiant qui a fait le stage
date_soutenance	Date	La date de la soutenance
Encadrant	Entier	Le numéro de l'enseignant qui a encadré le stage
Rapporteur	Entier	Le numéro de l'enseignant désigné comme rapporteur du
President	Entier	Le numéro de l'enseignant qui préside la soutenance
Decision	Chaine de caractères	La décision prise ('Valide' ou bien 'Non valide')
Note	Réel	La note de l'étudiant à la soutenance du stage

NB : On indique une valeur pour l'attribut encadrant seulement dans le cas d'un PFE.

## PARTIE I

(6 POINTS)

- 1) Donnez le modèle entité association qui correspond à ce schéma relationnel
  - 2) Insérez le stage suivant effectué par l'étudiant numéro 587 :
- |    |                  |           |            |     |
|----|------------------|-----------|------------|-----|
| 17 | Perfectionnement | Smart Lab | 01/02/2021 | 587 |
|----|------------------|-----------|------------|-----|
- 3) Une erreur s'est produite lors de la saisie des notes des soutenances des stages. Ecrivez la requête qui permet d'ajouter 1 point à la note de tous les étudiants qui ont soutenus leurs stages dans la période du 01/01/2020 au 31/03/2020.
  - 4) Supprimez les soutenances qui n'ont pas été programmées (aucune date de soutenance mentionnée)

## PARTIE 2

(8 POINTS)

Donnez les requêtes qui permettent de :

- 1) Afficher le nombre de stages auxquels on n'a pas encore affecté des encadrants.
- 2) Afficher les stages (code, type et date de début) effectués dans la même entreprise que l'étudiant numéro 412.
- 3) Afficher les détails des stages qui vont débuter le 1 février 2021 (code stage, type stage, nom de l'étudiant, prénom de l'étudiant et son groupe).
- 4) Afficher les noms et prénoms des enseignants qui ont assisté (en tant que rapporteur ou encadrant) à une soutenance de stage présidé par l'enseignant n°5.



## **EXAMEN**

Matière : Droit et PI

#### **Documents : Non autorisés**

Enseignante : Derouiche Hana

Durée : 1h30

Classe : DS121, DS122, DS123, MDW21, MDW22, RSI21, SEM21

Date : Janvier 2021

## **Partie I : Consultation**

### Exercice 1 :

- Monsieur Ahmed souhaite créer une entreprise dont l'activité sera de créer une boutique qui vendra des produits tels que vêtements, accessoires, articles en cuir, ...
  - Monsieur Ahmed voulait nommer sa boutique sous le nom « **Cloud Nine** » et le déposer dans les classes 18 (articles en cuir, sacs et valises), et 25 (vêtements).

Mais : le nom de domaine **cloudnine.fr** existe déjà et héberge une société de création de sites Internet depuis 2010, même si le nom de domaine n'a pas fait l'objet d'un enregistrement<sup>\*\*</sup> de marque.

Il existe aussi une société « **Cloud 10** » depuis 2008 en classe 25 pour les vêtements, mais leurs vêtements sont très différents, le design est différent et la clientèle plus haut de gamme.

1) Est-ce que Monsieur Ahmed peut déposer « Cloud Nine » dans les classes 18 et 25 ? Expliquez.

## Exercice 2 :

Monsieur Salah a élaboré un nouveau genre de logiciel qu'il a choisi de le dénommer « Windro », doublement compatible avec le système d'exploitation « Windows » et « Androïd ».

Après un profond travail de programmation et d'analyse, Monsieur Salah a finalement commercialisé son produit sous une marque dénommée « **Drown** » en choisissant l'enseigne d'un éléphant.

- 1) La création de monsieur Salah est, principalement, protégé par quel type de propriété ? Justifiez votre réponse ;
  - 2) Est-ce-que la création numérique de Monsieur Salah peut être protégée par un brevet d'invention ? Justifiez votre réponse ;

3) Est-il possible de protéger certains éléments de la création de monsieur Salah par la propriété industrielle ? Justifiez votre réponse ;

## **Partie I : Etude comparative**

Remplissez le tableau suivant :

Droit de Brevets	Droit des marques	Droit des dessins et modèles
Qu'est ce qu'un brevet ? ..... ..... ..... .....	Qu'est ce qu'une marque ? ..... ..... ..... .....	Qu'est ce qu'un dessin ou modèle industriel ? ..... ..... ..... .....
Conditions de protection : • ..... ..... ..... ..... ..... • ..... ..... ..... ..... ..... • ..... ..... ..... ..... .....	Conditions de protection : • ..... ..... ..... ..... ..... • ..... ..... ..... ..... ..... • ..... ..... ..... ..... .....	Conditions de protection : • ..... ..... ..... ..... ..... • ..... ..... ..... ..... ..... • ..... ..... ..... ..... .....
Durée de protection : .....	Durée de protection : .....	Durée de protection : .....
Frais d'enregistrement et de maintien en vigueur : ..... ..... ..... .....	Frais d'enregistrement et de maintien en vigueur : ..... ..... ..... .....	Frais d'enregistrement et de maintien en vigueur : ..... ..... ..... .....
Publication : Quand ? Où ? .....	Publication : Quand ? Où ? .....	Publication : Quand ? Où ? .....
Date d'opposition : .....	Date d'opposition : .....	Date d'opposition : .....
Sanctions : .....	Sanctions : .....	Sanctions : .....

- Une méthode `quiAppelle(Abonne contact, int d)` qui parcourt le tableau liste d'abonnés et retourne une chaîne avec les noms concaténés des abonnés qui peuvent appeler le contact indiqué pendant la durée d. Si aucun abonné de la liste ne peut le faire, elle lève l'exception `SoldeInsuffisantException` avec un paramètre égale à 1 puis retourne une chaîne vide.
- Une méthode `quiEnvoieSMS()` qui parcourt le tableau liste d'abonnés et retourne une chaîne avec les noms concaténés des abonnés qui peuvent envoyer un SMS. Si aucun abonné de la liste ne peut le faire, elle lève l'exception `SoldeInsuffisantException` avec un paramètre égale à 2 puis retourne une chaîne vide.
- Une méthode `main` où on crée un objet `friends` de la classe `Ami` dans lequel on crée et on ajoute les objets du tableau suivant :

Type d'abonné	Nom	Numéro	Solde
Orange	Sonia	55111222	45 millimes
Ooredoo	Jihene	22111333	60 millimes
Orange	Houda	55333222	710 millimes

Ensuite, on crée l'abonné encadrant et on essaie de le contacter, d'abord par SMS puis par appel téléphonique de **durée=5min**, en affichant les noms des abonnés dans liste qui peuvent le faire (utilisez les méthodes `quiEnvoieSMS`, `quiAppelle`). Au cas où personne ne peut contacter l'encadrant, on recharge le solde de tous les abonnés dans liste de **5 dinars**. Enfin, on identifie tous les abonnés de la liste.



## EXAMEN

Matière : PROGRAMMATION ORIENTEE OBJET

Enseignantes : Mejdoub S., Guerbouj S., Hedhli J., Mhafdh H.

Classes : DSI2, RSI2

Documents : Non autorisés

Durée : 1h30

Date: 18/01/2021

Le devoir comporte 7 pages

Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.

### EXERCICE 1

(4 POINTS)

1°/ Donner la trace d'exécution de cette classe.

```
public class test {
    public static void main (String[] args) {
        try {
            System.out.println("just trying");
            int value = Integer.parseInt ("140bb");
            System.out.println("end of try clause");
        } catch (NumberFormatException exception) {
            System.out.println("catching");
            System.out.println("End of catching");
        } finally {
            System.out.print ("finally clause");
            System.out.print ("end of the main");
        }
    }
}
```

2°/ Soient les classes Mere et Fille suivante :

```
1  public class Mere{
2  public String nom;
3  Mere(){}
4  Mere(String a){ nom=a; }
5  public void saluer(String n){
6      System.out.println("Bonjour "+n);}}
7  .
8  class Fille extends Mere {
9  public void f(){
10     System.out.println("bonsoir "+nom); }
11  public static void main(String[] args){
12     Fille f=new Fille();
13     f.saluer("java") ; }}
```

Ce bout de code :

- S'exécute correctement et affiche bonjour java
- S'exécute correctement et affiche bonsoir
- Génère une erreur de compilation à cause de la ligne 9
- Génère une erreur de compilation à cause de la ligne 11

**EXERCICE 2****(16 POINTS)**

Des étudiants n'ont pas beaucoup de solde dans leurs téléphones alors qu'ils ont besoin de contacter leurs encadrant. Ils ont développé une petite application en Java pour déterminer lequel d'eux peut appeler cette personne pour une durée spécifiée.

1°/ Soit la classe abstraite **Abonne** qui possède les caractéristiques suivantes :

- **num**: un entier qui représente le numéro de téléphone de l'abonné.
- **solde** : un réel qui représente le solde en dinar.
- **nom** : une chaîne contenant le nom de l'abonné.
- **getOperateur()** : une méthode abstraite qui retourne une chaîne contenant le nom de l'opérateur de l'abonné.
- **getNom()** : un accesseur qui retourne le nom de l'abonné.
- **getSolde()** : un accesseur qui retourne le solde d'un abonné.
- **rechargerSolde(double m)** : une méthode qui ajoute au solde le montant m.
- **identifier()** : une méthode qui affiche tous les attributs d'un abonné.

La classe **Abonne** implémente l'interface **Communication** suivante :

```
public interface Communication {  
    public boolean envoyerSMS();  
    public boolean appeler(Abonne contact, int duree);  
}
```

L'interface contient ces méthodes :

- **envoyerSMS()** : retourne faux si le solde ne permet pas d'envoyer un SMS (sachant que le cout d'envoi d'un SMS est 60 millimes). Si le solde est suffisant, alors, le solde est diminué du cout d'un SMS puis la méthode retourne vrai.
- **appeler (Abonne contact, int duree)** : calcule d'abord le cout d'appel à *contact* pendant la *durée* donnée, puis, retourne Faux si le solde est insuffisant pour passer l'appel. Si le solde est suffisant, alors, elle diminue le solde d'un montant égale au cout de l'appel et retourne Vrai.

**NB** : Le cout de l'appel est calculé à partir de la formule suivante :

$$Cout = tarif \ minute \times \ durée$$

Sachant que le tarif d'une minute vaut 90 millimes si le contact appelé est du même opérateur sinon le tarif vaut le double (180 millimes).

Donnez le code de la classe **Abonne**.

2°/ Soit la classe **AbonneOrange** qui hérite de la classe **Abonne** et définit les méthodes suivantes :

- un constructeur : initialise tous les attributs hérités de la classe.
- **getOperateur()** : retourne « Orange » si le numéro de l'abonné commence par 5, sinon, elle retourne « Inconnu ». (Astuce : Utilisez la division)

Donnez le code de la classe **AbonneOoredoo**.

On suppose qu'on vous donne aussi une classe **AbonneOoredoo** qui hérite de la classe **Abonne** et définit les méthodes suivantes :

- un constructeur : initialise tous les attributs hérités de la classe.
- **getOperateur()** : retourne « Ooredoo » si le numéro de l'abonné commence par 2, sinon, elle retourne « Inconnu ».

N'écrivez pas la classe AbonneOoredoo.

3°/ Soit une classe d'exception intitulée **SoldeInsuffisantException** qui possède un constructeur avec un paramètre **choix** de type entier.

- Si le choix vaut 1, le constructeur affiche "Soldes insuffisants pour passer cet appel ! Veuillez recharger vos soldes".
- Si le choix vaut 2, le constructeur affiche "Soldes insuffisants pour envoyer un message ! Veuillez recharger vos soldes".

Donnez le code de la classe **SoldeInsuffisantException**.

4°/ Complétez le code de la classe **Amis** (voir pages 5, 6, 7) sachant qu'elle possède :

- Un attribut **liste** : qui représente un tableau d'**Abonne**.
- Un attribut **nbAbonnes** : un entier initialisé à 0 et qui représente le nombre effectif d'abonnés dans le tableau **liste**.
- Un constructeur : qui crée le tableau **liste** avec une taille maximale donnée en paramètre.
- Une méthode **identifierTous()** : qui affiche tous les abonnés du tableau d'objets « **liste** ».
- Une méthode **rechargerTous(double m)** : recharge les soldes de tous les abonnés dans le tableau **liste** d'un montant m.
- Une méthode **rejoindreAmis(Abonne a)** : qui ajoute l'abonne « **a** » dans le tableau des abonnes « **liste** » si la taille maximale n'est pas atteinte.

EXAMEN DE : .....

\*\*\*\*\*NB : N'écrivez pas votre nom ici \*\*\*\*\*

## = Feuille à rendre

public class Amis {

.....  
.....

```
public Amis(int Nmax) {  
    liste = new Abonne[Nmax];  
}
```

```
public void rejoindreAmis(Abonne a) {  
    if (nbAbonnes < liste.length) {  
        liste[nbAbonnes] = a;  
        nbAbonnes++;  
    }  
    else  
        System.out.println("Liste pleine");  
}
```

```
public String quiAppelle(Abonne a, int d) {
```

String nomAmi="" .

```
for (int i=0; i<nbAbonnés; i++) {
```

3

```
if (nomAmi==")")
```

```
.....  
return nomAmi;
```

EXAMEN DE : .....

\*\*\*\*\*NB : N'écrivez pas votre nom ici \*\*\*\*\*

EXAMEN DE : .....

\*\*\*\*\*NB : N'écrivez pas votre nom ici \*\*\*\*\*

```
public String quiEnvoiSMS(Abonne a) ..... {  
    String nomAmi="";  
    for (int i=0; i<nbAbonnes; i++) {  
        if (.....)  
            .....  
    }  
    if (nomAmi=="")  
        .....  
    return nomAmi;  
}
```

```
public void rechargerTous(double m) {  
    .....  
}  
public void identifierTous() {  
    .....  
}
```

```
public static void main(String[] args) {  
    Amis friends = new Amis(5);  
    Abonne a = .....  
    friends.rejoindreAmis(a);
```

a = .....

friends.rejoindreAmis(a);

a = .....

friends.rejoindreAmis(a);

System.out.print("L'encadrant à contacter :");

a = .....

a.identifie();

try {  
 .....  
} catch (...) {  
 .....  
}  
finally {  
 .....  
}

```
}
```