

Exercice 1 (4 pts):

1. Le JRE est-il un ensemble d'outils permettant de développer des programmes Java ?
2. Quelle est la différence entre une variable de classe et une variable d'instance ?
3. Combien d'instances de la classe A crée le code suivant ?

A x, u, v; x = new A(); A y = x; A z = new A();

4. Considérons code ci-contre :

- Numéroter les lignes du code.
- Identifier les erreurs du code en indiquant le numéro de la ligne et l'erreur apparue.
- Corriger les erreurs.

```
Classe ProgrammeSimple{  
    public static void main(String[] args){  
        Scanner sc =new scanner();  
        System.out.println("Saisissez un entier : ");  
        int d = sc.next();  
        System.out.println("Vous avez saisi  
                                un entier d = " + d);  
    }  
}
```

Exercice 2 (6 pts):

Écrire un programme permettant de réaliser les opérations suivantes :

1. Déclarer un tableau de n entiers et affiche un message indiquant si les éléments sont triés dans un ordre croissant ou non.
2. Inverser l'ordre des éléments du tableau (sans utiliser de deuxième tableau).

Exercice 3 (10 pts):

Nous souhaitons développer une application Java permettant de gérer une **agence de location de voitures et de logements**. Un **logement** est caractérisé par *code*, *adresse*, *surface*, *nbPieces*, *prixParJour*, *nbJours*, *disponible*. Une **voiture** est caractérisée par *matricule*, *modèle*, *kilometrage*, *kilomDernierEntretien*, *prixParHeur*, *nbHeurs*, *disponible*. Les voitures de l'agence sont classées en **deux catégories** : **VoitureThermique** et **VoitureElectrique**.

1. Écrire la classe Logement en ajoutant un constructeur qui initialise les attributs : *code*, *adresse*, *surface*, *nbPieces*, *prixParJour*.
2. Écrire la classe Voiture de sorte qu'elle oblige ses classes descendantes à implémenter les méthodes suivantes : *afficher()*, *verifierBesoinEntretien()*.
3. Définir les classes VoitureThermique et VoitureElectrique qui sont des sous-classes de Voiture, de sorte que :
 - VoitureThermique possède un attribut type (essence ou diesel).
 - VoitureElectrique possède un attribut etatBatterie de type int.
 - Les deux classes possèdent un attribut constant kilMaintenance qui indique le nombre de kilomètres à parcourir avant de devoir effectuer un entretien.
 - Ajouter les accesseurs (getter et setter) nécessaires pour l'attribut *etatBatterie*.
 - Définir les méthodes *afficher()* et *verifierBesoinEntretien()*.
4. Nous souhaitons obliger les voitures et les logements à fournir une méthode nommée *louer()* : pour une voiture, le montant est calculé en fonction du nombre d'heures de location ; pour un logement, le montant est calculé en fonction du nombre de jours de location. Écrire le code nécessaire qui garantit cette contrainte. **Indication : Voiture et Logement peuvent hériter d'une classe abstraite ou implémente une interface offrant cette méthode**
5. À la fin de chaque journée, l'administrateur de l'agence souhaite recenser toutes les voitures actuellement en location (non disponibles). En supposant que l'agence dispose de 50 voitures stockées dans un tableau **tabV**. Écrire le code nécessaire pour afficher les informations de chaque voiture en location ainsi que le nombre total de voitures en location par catégorie.