

Nom:

Prénom:

Matricule:

EXAMEN

**INFORMATIQUE POUR LA GESTION II
(INFO-D-201)**

**2ème Année du Bachelier
Ingénieur de Gestion**

10 Janvier 2008

RESULTATS :

Q1 : Diagramme UML	/3
Q2 : UML → Code Java	/2
Q3 : Compréhension Java	/3
Q4 : Correction Code Java	/2
TOTAL DE L'EXAMEN	/10
<hr/>	
COTE SEMAINE INTENSIVE	/10
<hr/>	
TOTAL SECONDE SESSION	/20

Nom:

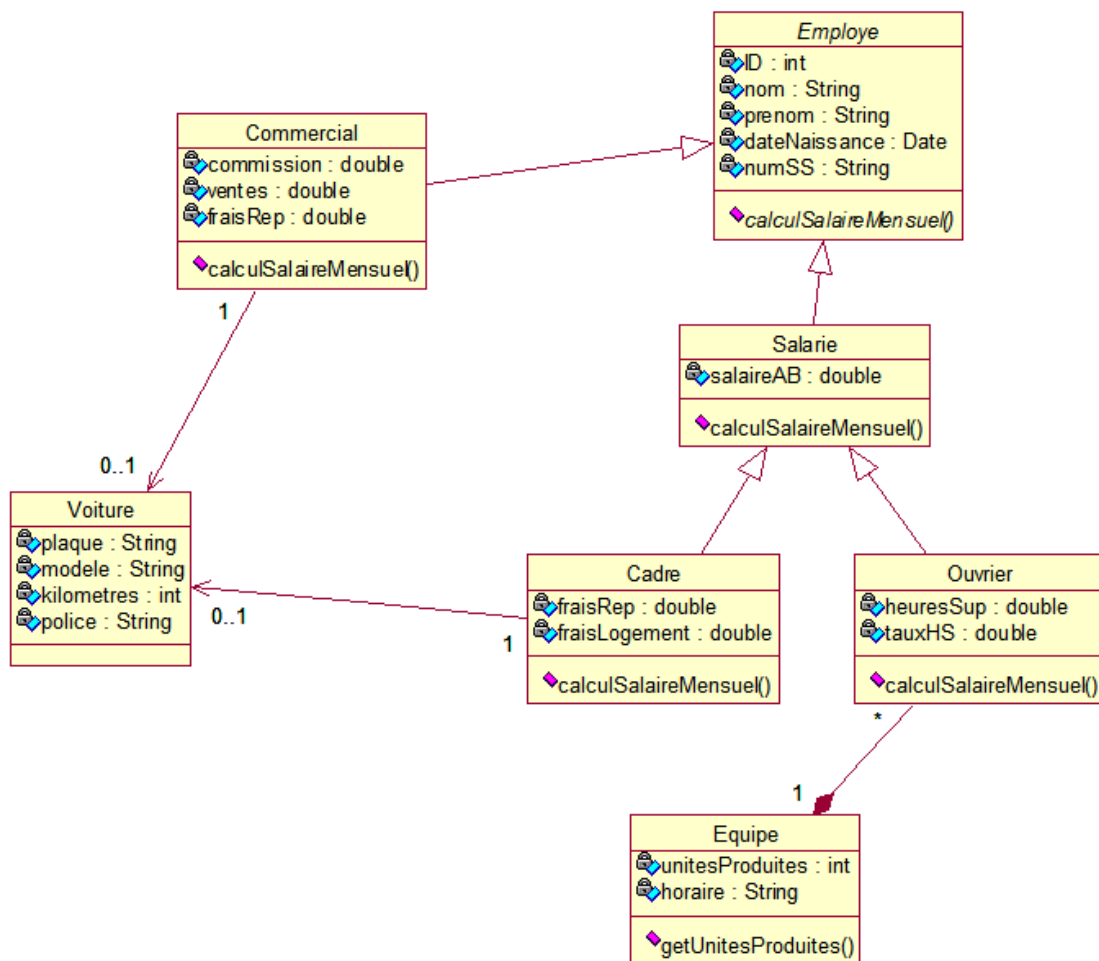
Prénom:

Matricule:

Question n°1: UML (/3)

Vous devez réaliser une application OO pour le département RH d’une société manufacturière. Tous les employés sont identifiés par un numéro ainsi que par leurs nom, prénom, date de naissance et numéro de sécurité sociale, et tous perçoivent une certaine forme de rémunération : les salariés perçoivent un salaire annuel brut fixe tandis que les commerciaux touchent un certain pourcentage (propre à chacun d’eux) des ventes qu’ils ont réalisées. Parmi les salariés, les ouvriers – à la différence des cadres – touchent une compensation pour leurs heures supplémentaires à un tarif horaire propre à chacun. Par ailleurs, les commerciaux et cadres bénéficient d’une indemnité forfaitaire pour leurs frais de représentation, et peuvent utiliser une voiture de fonction si nécessaire dont le système devra indiquer la plaque, le modèle, le kilométrage et le numéro de la police d’assurances. Les cadres ont en outre droit à une indemnité forfaitaire pour leur logement. Enfin, les ouvriers sont organisés en équipes dont les horaires sont établis à l’avance pour toute l’année. En fin d’année, chaque ouvrier perçoit une prime sur résultats au prorata du nombre d’articles produits par son équipe. L’application devra être capable de calculer le salaire mensuel de chaque employé quel qu’il soit. Veuillez tracer le diagramme de classes de l’application.

REPONSE :



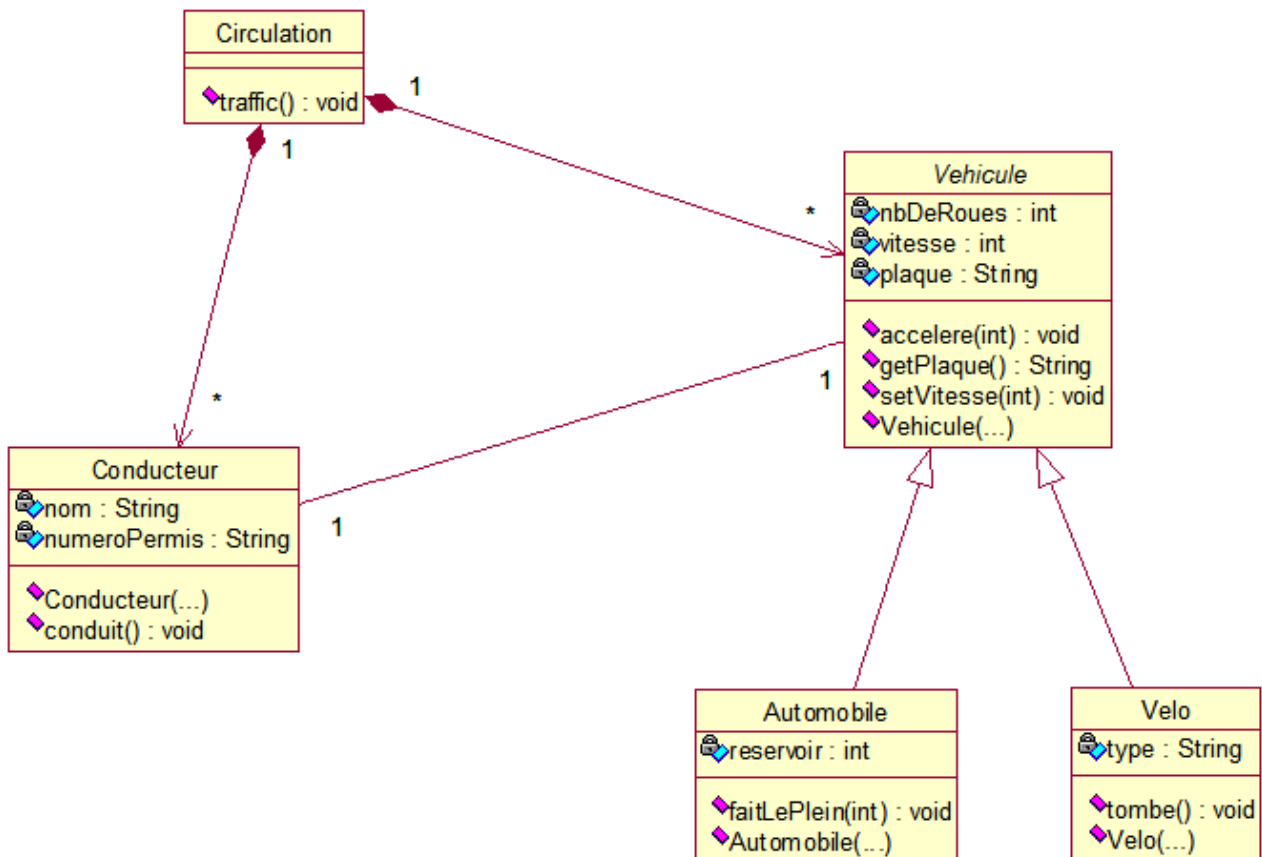
Nom:

Prénom:

Matricule:

Question n°2: UML → Code Java (/2)

Veillez produire le code Java minimum (sans implémenter les méthodes) correspondant au diagramme de classes ci-dessous. Votre code ne devrait pas « réaliser » quelque chose (puisque les méthodes ne seront pas implémentées), mais il devrait au minimum pouvoir être compilé sans erreur.

**REPONSE :**

```

public class Circulation {
    private ArrayList<Vehicule> vehicules;
    private ArrayList<Conducteur> conducteurs;
    public void traffic(){
    }
}

```

Nom:**Prénom:****Matricule:**

REPONSE (SUITE):

```
public class Conducteur {
    private String nom;
    private String numeroPermis;
    private Vehicule monVehicule;
    public Conducteur(String n, String p) {
        this.nom = n;
        this.numeroPermis = p;
    }
    public void conduit() {
    }
}

public class Vehicule {
    private Conducteur leConducteur;
    private int nbDeRoues;
    private int vitesse;
    private String plaque;
    public Vehicule(int r, int v, String p, Conducteur c) {
        nbDeRoues = r;
        vitesse = v;
        this.plaque = p;
        leConducteur = c;
    }
    public void accelere(int dv) {
    }
    public String getPlaque() {
        return plaque;
    }
    public void setVitesse(int vitesse) {
        this.vitesse = vitesse;
    }
}

public class Automobile extends Vehicule {
    private int reservoir;
    public Automobile(int v, int r, String p, Conducteur c) {
        super(4, v, p, c);
        reservoir = r;
    }
    public void faitLePlein(int quantite) {
    }
}

public class Velo extends Vehicule {
    private String type;
    public Velo(int v, String t, String p, Conducteur c) {
        super(2, v, p, c);
        this.type = t;
    }
    public void tombe() {
    }
}
```

Nom:**Prénom:****Matricule:****Question n°3: Compréhension Code Java (/3)**

Lisez attentivement le code Java ci-dessous (garanti sans erreur). Que se passe-t-il à l'exécution (i.e. que va-t-il apparaître à l'écran) ?

```

public abstract class Livre {
    private String titre = null;
    private int nbCopies = 0;
    private String publicVise = "tout public";
    public Livre(String t, String publicVise) {
        titre = t;
        this.publicVise = publicVise;
    }
    public String getTitre(){
        return titre;
    }
    public boolean jeSuisPourAdulte(){
        if(this.publicVise == "tout public")
            return false;
        return true;
    }
    public void ajouterCopies(int c){
        nbCopies+=c;
    }
    protected void supprimerCopies(int c){
        nbCopies-=c;
    }
    protected int getNbCopies(){
        return nbCopies;
    }
    public void Print(){
        System.out.println("Livre "+titre+ " - "+publicVise+ " - "+nbCopies+ " copie(s).");
    }
}

public class LivreAVendre extends Livre {
    private int prixVente;
    public LivreAVendre(String t, String pub, int p) {
        super(t, pub);
        this.prixVente = p;
    }
    public int getPrixVente(){
        return prixVente;
    }
    public void vendreCopies(int copiesVendues){
        if(copiesVendues<=this.getNbCopies())
            this.supprimerCopies(copiesVendues);
        else
            System.out.println("Plus assez d'exemplaires en stock");
    }
    public void Print(){
        super.Print();
        System.out.println("\tPrix de vente: "+prixVente+ " euro");
    }
}

```

Nom:**Prénom:****Matricule:**

```

public class LivreALouer extends Livre {
    private double prixLocation = 10.0;
    private int locationsMaxAutorisees = 0;
    private java.util.ArrayList<Exemplaire> copies;
    public LivreALouer(String t, double p, String pub, int l) {
        super(t, pub);
        prixLocation = p;
        locationsMaxAutorisees = l;
        copies = new java.util.ArrayList<Exemplaire>();
    }
    public void ajouterCopies(int c){
        for(int i=0;i<c;i++){
            super.ajouterCopies(1);
            copies.add(new Exemplaire(getNbCopies(),0,"disponible"));
        }
    }
    public int loueLeLivre(){
        for(int i=0;i<copies.size();i++){
            if(copies.get(i).loueMoi(this.locationsMaxAutorisees))
                return copies.get(i).getNumeroID();
        }
        System.out.println("Pas de copie disponible pour l'instant!");
        return -1;
    }
    public void retourLivre(int numeroExemplaire){
        for(int i=0;i<copies.size();i++){
            if(copies.get(i).getNumeroID()==numeroExemplaire)
                copies.get(i).retourCopie();
        }
    }
    public void augmenteLePrix(int pourcentage){
        double d = ((double) pourcentage)/100;
        this.prixLocation *= (1+d);
    }
    protected int getNbCopiesDispo(){
        int n = 0;
        for(int i=0;i<copies.size();i++)
            if(copies.get(i).esTuDisponible())
                n++;
        return n;
    }
    public double getPrixLocation(){
        return this.prixLocation;
    }
    public void Print(){
        super.Print();
        System.out.println("\tPrix de location: "+prixLocation+" euro
        - louable "+locationsMaxAutorisees+" fois - " +
        getNbCopiesDispo() + " copie(s) disponible(s)");
        for(int i=0;i<copies.size();i++)
            copies.get(i).Print();
    }
}

```

Nom:**Prénom:****Matricule:**

```

public class Exempleire {
    private int numeroID;
    private int nbLocations;
    private String statut;
    public Exempleire(int id, int nbLoc, String s){
        numeroID = id;
        nbLocations = nbLoc;
        statut = s;
    }
    public boolean esTuDisponible(){
        if(statut=="disponible")
            return true;
        return false;
    }
    public boolean loueMoi(int maxLoc){
        if(this.esTuDisponible() && this.nbLocations<maxLoc){
            this.statut = "en location";
            this.nbLocations++;
            return true;
        }
        return false;
    }
    public void retourCopie(){
        this.statut = "disponible";
    }
    public void Print(){
        System.out.println("\t> Ex. n° "+numeroID+" actuellement "+statut+
            " - loué "+nbLocations+" fois");
    }
    public int getNumeroID() {
        return numeroID;
    }
}

```

```

public class Bibliotheque {
    private java.util.ArrayList<Livre> livres;
    public Bibliotheque(){
        livres = new java.util.ArrayList<Livre>();
    }
    public void job(){
        LivreALouer l1 = new LivreALouer("Harry Butter et les
            Oméga3",10,"tout public",50);
        livres.add(l1);
        LivreALouer l2 = new LivreALouer("Programmer en Java en
            s'amusant",8.5,"pour adultes",5);
        livres.add(l2);
        LivreAVendre l3 = new LivreAVendre("L'orienté objet","tout
            public",35);
        livres.add(l3);
        for(int i=0;i<livres.size();i++)
            livres.get(i).ajouterCopies(i+1);
        l1.loueLeLivre();
        l1.augmenteLePrix(5);
        l3.vendreCopies(2);
        for(int i=0;i<2;i++){
            if(l2.loueLeLivre(>0){
                System.out.println("Waouw, quel bon livre!");
            }
        }
    }
}

```

Nom:

Prénom:

Matricule:

```
    }
    int j = l2.loueLeLivre();
    if(j>0){
        System.out.println("Pff, n'importe quoi ce bouquin!");
        l2.retourLivre(j);
    }
}
l3.vendreCopies(15);
l2.retourLivre(1);
for(int i=0;i<livres.size();i++)
    livres.get(i).Print();
}
public static void main(String[] args) {
    Bibliotheque NB = new Bibliotheque();
    NB.job();
}
}
```

REPONSE :

Waouw, quel bon livre!
Pff, n'importe quoi ce bouquin!
Waouw, quel bon livre!
Pas de copie disponible pour l'instant!
Plus assez d'exemplaires en stock
Livre Harry Butter et les Oméga3 - tout public - 1 copie(s).
Prix de location: 10.5 euro - louable 50 fois - 0 copie(s) disponible(s)
> Ex. n° 1 actuellement en location - loué 1 fois
Livre Programmer en Java en s'amusant - pour adultes - 2 copie(s).
Prix de location: 8.5 euro - louable 5 fois - 1 copie(s) disponible(s)
> Ex. n° 1 actuellement disponible - loué 1 fois
> Ex. n° 2 actuellement en location - loué 2 fois
Livre L'orienté objet - tout public - 1 copie(s).
Prix de vente: 35 euro

Nom:**Prénom:****Matricule:****Question n°4: Correction Code Java (/2)**

Le code Java ci-dessous contient quelques erreurs de logique qui seraient détectées par le compilateur. Veuillez identifier, corriger et expliquer brièvement ces erreurs dans le code directement.

ATTENTION :

La justification importe autant que l'identification et la correction des erreurs.
Noter des erreurs partout à tort et à travers pourrait être pénalisé !

```

public abstract class Arbre {
    protected int x, y, hauteur, diametre, branches, terminaisons;
    private String etat;
    public Arbre(int x,int y,int h,int d,int b,String e,String t) {
        this.x = x;
        this.y = y;
        this.hauteur = h;
        this.diametre = d;
        this.branches = b;
        this.terminaisons = 0;
        this.etat = etat;           Aurait du être this.etat=e
        this.type = type;         Aurait du être this.type=t
    }
    private void afficheToi(){
        System.out.println(type+"\t"+x+"\t"+y+"\t"+hauteur+"\t"+
            diametre+"\t"+branches+"\t"+terminaisons+"\t"+etat);
        this.afficheTerminaisons();
    }
    public abstract void afficheTerminaisons();
    public abstract void meursEnAutomne();
    public void changeEtat(String s){
        etat = s;
    }
    public void pousser(){
        this.branches++;
        this.hauteur+=2;
        this.diametre++;
    }
}

public class Conifere extends Arbre {
    private java.util.ArrayList<Epine> lesEpines;
    public Conifere() {           Manque appel au super constructeur
        lesEpines = new java.util.ArrayList<Epine>();
    }
    public void meursEnAutomne() {
        for(int i=0;i<lesEpines.size();i++)
            lesEpines.get(i).changeDeCouleur("Jaune");
        this.changeEtat(false); Méthode changeEtat attend un String pas un boolean
    }
    public void pousser() {
        super.pousser();
        for(int i=0;i<this.branches;i++){
            lesEpines.add(new Epine(5,"Vert"));
            this.terminaisons++;
        }
    }
}

```

Nom:

Prénom:

Matricule:

```

    public void afficheTerminaisons(){
        for(int i=0;i<lesEpines.size();i++){
            lesEpines.get(i).afficheToi();
        }
    }
}
public class Feuillus extends Arbre {
    private java.util.ArrayList<Feuille> lesFeuilles;
    public Feuillus() { Manque appel au super constructeur
        lesFeuilles = new java.util.ArrayList<Feuille>();
    }
    public void meursEnAutomne() {
        lesFeuilles.clear();
        this.changeEtat("Dénudé");
    }
    public void pousser() {
        super.pousser();
        for(int i=0;i<lesFeuilles.size();i++){
            lesFeuilles.get(i).pousser();
        }
        lesFeuilles.add(new Feuille(3,"Vert"));
        this.terminaisons++;
    }
    public void afficheTerminaisons(){
        for(int i=0;i<lesFeuilles.size();i++){
            lesFeuilles.get(i).afficheToi();
        }
    }
    public void changeDeCouleur(String c){
        for(int i=0;i<lesFeuilles.size();i++){
            lesFeuilles.get(i).changeDeCouleur(c);
        }
    }
}
public abstract class Terminaison {
    private String couleur;
    public Terminaison(String c) {
        this.couleur = c;
    }
    public Terminaison(){
        this("Vert");
    }
    public void changeDeCouleur(String c){
        this.couleur = c;
    }
    public abstract void afficheToi();
}
public class Epine extends Terminaison {
    private int longueur;
    public Epine(int l, String c) {
        this.longueur = l;
    }
    public void afficheToi(){
        System.out.println("\tE: "+longueur+"cm - "+this.couleur);
        Attribut couleur est privé dans classe parent Terminaison
    }
}

```

Nom:

Prénom:

Matricule:

```

public class Feuille extends Terminaison {
    private int diametre;
    public Feuille(int d, String c) {
        this.diametre = d;
    }
    public void afficheToi(){
        System.out.println("\tF: "+diametre+"cm");
    }
    public void pousser(){
        this.diametre++;
    }
}
public class Foret {
    private java.util.ArrayList<Arbre> lesArbres;
    public Foret(){
        lesArbres = new java.util.ArrayList<Arbre>();
        lesArbres.add(new Feuillus(15,15,7,3,2,"En pleine forme","Chêne"));
        lesArbres.add(new Conifere(100,20,5,2,1,"En pleine forme","Sapin"));
        lesArbres.add(new Conifere(200,20,10,2,"En pleine forme","Pin"));
        Constructeur Conifere(int,int,int,int,String,String) invalide
        lesArbres.add(new Feuillus(250,15,5,2,1,"En pleine forme","Hêtre"));
        lesArbres.add(new Arbre(10,30,3,2,1,"En pleine forme","Eglantier"));
        Classe Arbre étant abstraite, elle ne peut être instanciée
        lesArbres.add(new Conifere(350,20,3,1,0,"En pleine forme","Pin"));
        lesArbres.add(new Feuillus(40,15,4,2,1,"En pleine forme","Bouleau"));
        lesArbres.get(0).changeDeCouleur("Rouge");
        Méthode changeDeCouleur(String) n'existe pas dans Arbre
    }
    public void lesArbresPoussent(){
        for(int i=0;i<lesArbres.size();i++)
            lesArbres.get(i).pousser();
    }
    public void lesArbresSAffichent(){
        System.out.println("\nTYPE\tX\tY\tH\tD\tB\tN\tETAT");
        for(int i=0;i<lesArbres.size();i++)
            lesArbres.get(i).afficheToi();
        Méthode afficheToi() est privée dans Arbre
    }
    public void lesArbresMeurentEnAutomne(){
        for(int i=0;i<lesArbres.size();i++)
            lesArbres.get(i).meursEnAutomne();
    }
    public static void main(String[] args){
        monBeauSapin = new Conifere(20,20,10,2,1,"En pleine forme","Pin")
        Variable non déclarée
        Foret f = new Foret();
        for(int i=0;i<5;i++){
            f.lesArbresPoussent();
        }
        f.lesArbresMeurentEnAutomne();
        lesArbresSAffichent(); Sur quoi s'exécute la méthode ?
    }
}

```