Nom :
Prénom :
Numéro d'étudiant :

Contrôle 2 - Classes, héritage, ArrayList

Question 1. Savoir utiliser une classe déjà définie. Etudiez le listing de la classe `Concert` (listing 1).

Listing 1 – Concert.java

```
package controle;
public class Concert {
    private String artiste;

    // a la creation du concert, pas encore de participants
    // cet attribut ne doit pas etre passe en parametre a un constructeur
    private int nbParticipants=0;

    private double prixParPersonne;

    // le prix de l'assurance est le meme pour tous les participants
    private static double prixAssurancePersonnelle = 5;

    public Concert() {}

    public Concert(String artiste, double prixParPersonne) {
        this.artiste = artiste;
        this.prixParPersonne = prixParPersonne;  }

    public String getArtiste() {return artiste;}

    public void setArtiste(String artiste) {this.artiste = artiste;}

    public int getNbParticipants() {return nbParticipants;}

    public void setNbParticipants(int nbParticipants)
        {this.nbParticipants = nbParticipants;}

    public double getPrixParPersonne() {return prixParPersonne;}

    public void setPrixParPersonne(double prixParPersonne)
        {this.prixParPersonne = prixParPersonne;}

    public static double getPrixAssurancePersonnelle()
        {return prixAssurancePersonnelle;}

    public static void setPrixAssurancePersonnelle
                    (double prixAssurancePersonnelle)
        {Concert.prixAssurancePersonnelle = prixAssurancePersonnelle;}

    public String toString() {
        return "Concert [artiste=" + artiste + ", nbParticipants="
                + nbParticipants + ", prixParPersonne=" + prixParPersonne + "]";
    }

    // la methode d'inscription augmente le nombre de participants
    // et retourne le prix de l'inscription
    public double inscritEtDonnePrix(int nbPersonnesAInscrire){
        nbParticipants+=nbPersonnesAInscrire;
        return nbPersonnesAInscrire*(prixParPersonne
                    +Concert.prixAssurancePersonnelle);
    }
```

Remplissez le main du listing 2 afin qu'il effectue les opérations suivantes :

1. Créer une instance de `Concert` pour l'artiste *Didier Lockwood* avec un prix par personne égal à 15.
2. Inscrire 4 personnes et afficher le prix résultant, retourné par la méthode d'inscription.

Listing 2 – Programme.java

```
package controle;
public class ProgrammeConcert {
    public static void main(String[] args){









    }
}
```

Question 2. Savoir écrire le code d'une sous-classe

a- Ecrire l'entête et l'attribut d'une classe représentant les *concerts en salle* avec les informations suivantes. Un concert en salle est une sorte de concert muni de plus d'un attribut représentant la capacité de la salle en nombre de participants (cette capacité, représentant le nombre maximum de personnes susceptibles d'être accueillies est propre à chaque concert en salle).

Réponse à la question 2.a :

b- Ecrire pour la classe représentant les *concerts en salle* un constructeur prenant tous les paramètres nécessaires pour initialiser les attributs d'instance qui doivent l'être.

Réponse à la question 2.b :

c- Redéfinir (en la spécialisant) dans la classe représentant les *concerts en salle* la méthode `inscritEtDonnePrix` de manière à n'inscrire effectivement les personnes que s'il y a encore assez de places pour tout le groupe que l'on cherche à inscrire. Par exemple, si on cherche à inscrire 4 personnes en même temps (c'est la valeur passée en paramètre à la méthode) et qu'il reste 2 places seulement, on n'inscrit personne. Dans ce type de cas, c'est-à-dire lorsqu'il n'y a plus assez de place pour tout le groupe, on affiche un message d'erreur et on retourne 0.

Réponse à la question 2.c :

Question 3. Savoir manipuler une liste d'objets

a- Ecrire l'entête et les attributs d'une classe `Festival` avec les informations suivantes. Un festival a un nom et se compose de plusieurs concerts. Initialisez les deux attributs au moment de leur déclaration.

Réponse à la question 3.a :

b- Ecrire, pour la classe `Festival`, une méthode permettant d'ajouter, à la liste des concerts, un concert qui n'y apparaît pas (si le concert apparaît déjà dans la liste, la méthode ne fait rien).

Réponse à la question 3.b :

c- Ecrire, pour la classe `Festival` , une méthode calculant et retournant le nombre de places vendues dans tous les concerts (somme des nombres de participants de chacun des concerts).

Réponse à la question 3.c :

d- Ecrire, pour la classe `Festival`, une méthode retournant les concerts ayant plus de 10 participants.

Réponse à la question 3.d :

e- Ecrire un programme où l'on crée deux concerts (dont un concert en salle), un festival, puis on ajoute les concerts au festival et enfin on affiche le résultat des deux méthodes définies sur `Festival` (nombre de places vendues et concerts ayant plus de 10 participants).

Réponse à la question 3.e :