

Contrôle continu optionnel HAI7171 – CC4 : janvier 2023
Programmation par objets
(1h15)

Université de Montpellier – Faculté Des Sciences
Master informatique (ICO), géomatique, bioinformatique, Physique numérique

Nous étudions les éléments d'un logiciel de gestion de cours de gymnastique et de cartes d'achats de cours pour un club. Nous vous donnons pour commencer une énumération représentant les types de lieux où peut avoir le cours et une partie de la classe représentant les cours de gymnastique.

```
public enum TypeLieu {stade, salle}
```

```
public class CoursGym {  
    private int dureeMn; // duree en minutes  
    private boolean tapisSolRequis = false; // true si un tapis est necessaire  
    private TypeLieu lieu;  
  
    public CoursGym() {}  
  
    public CoursGym(int dureeMn, boolean tapisSolRequis, TypeLieu lieu) {  
        this.setDureeMn(dureeMn);  
        this.setTapisSolRequis(tapisSolRequis);  
        this.setLieu(lieu);  
    }  
  
    public int getDureeMn() {return this.dureeMn;}  
    public void setDureeMn(int dureeMn) {this.dureeMn = dureeMn;}  
  
    public boolean isTapisSolRequis() {return this.tapisSolRequis;}  
    public void setTapisSolRequis(boolean tapisSolRequis)  
    {this.tapisSolRequis = tapisSolRequis;}  
  
    public TypeLieu getLieu() {return this.lieu;}  
    public void setLieu(TypeLieu lieu) {this.lieu = lieu;}  
}
```

Question 1 (2 pts). Réécrivez la méthode **setDureeMn** de manière à ce que la durée d'un cours ne puisse être que de 30mn, de 60mn ou de 90mn.

```
public void setDureeMn(int dureeMn) {  
    if (dureeMn==30 || dureeMn==60 || dureeMn==90) // 1  
        {this.dureeMn = dureeMn;} //0,5  
    else {System.out.println("Cette durée de cours n'est pas autorisée");} //0,5  
}
```

Question 2 (2 pts). Écrivez une méthode calculant et retournant le prix d'un cours. Ce prix est de 5 euros si la durée est 30mn, 9 euros si la durée est 60mn et 12 euros si la durée est 90mn, sinon un message d'erreur est affiché et la valeur -1 est retournée.

```
public double prix() { //0,5
    if (this.dureeMn==30) return 5; //0,25
    if (this.dureeMn==60) return 9; //0,25
    if (this.dureeMn ==90) return 12; //0,25
    else { //0,25
        System.out.println("Cette durée de cours n'est pas autorisée"); //0,25
        return -1; //0,25
    }
}
```

Question 3 (2 points). Écrivez les trois instructions d'un programme **main** pour (1) créer un cours de gym en salle de 60mn et demandant un tapis de sol, (2) changer la durée en 90mn, (3), afficher son prix.

```
CoursGym cg1 = new CoursGym(60,true,TypeLieu.salle); //0,75
cg1.setDureeMn(90); //0,5
System.out.println(cg1.prix()); //0,75
```

On vous donne à présent le début de la classe représentant les cartes d'achat de cours. Une carte d'achat contient une liste de cours.

```
public class CarteAchatCours {
    private ArrayList<CoursGym> listeCours = new ArrayList<>();
    public CarteAchatCours() {}
    public void ajoute(CoursGym cg) {this.listeCours.add(cg);}
}
```

Question 4 (2 points). Écrivez une méthode calculant et retournant le prix à payer pour les cours de la carte.

```
public double prixAPayer() { //0,25
    double p=0; //0,5
    for (CoursGym cg : this.listeCours) //0,5
        p+=cg.prix(); //0,5
    return p; //0,25
}
```

Question 5 (3 points). Écrivez une méthode retournant la sous-liste des cours de la carte correspondant à une certaine durée passée en paramètre.

```
public ArrayList<CoursGym> sousListeCoursDuree(int d){ //0,5
    ArrayList<CoursGym> res = new ArrayList<>(); //0,5
    for (CoursGym cg : this.listeCours) //0,5
        if (cg.getDureeMn()==d) //0,5
            res.add(cg); //0,5
    return res; //0,5
}
```

Question 6 (2 points). Écrivez les instructions de la suite du programme main pour (1) créer une carte d'achat, (2) y ajouter le cours créé à la question 3 et (3) afficher le prix de la carte.

```
CarteAchatCours carte = new CarteAchatCours(); //0,75
carte.ajoute(cg1); //0,5
System.out.println(carte.prixAPayer()); //0,75
```

Question 7 (7 points). Nous vous donnons maintenant une classe représentant les accessoires. Un accessoire peut être un élastique, un ballon, un poids, etc. Il a une dénomination et un prix de location pour 30mn.

```
public class Accessoire {
    private String denomination;
    private double prixLocation30mn;
    public Accessoire() {}
    public Accessoire(String denomination, double prixLocation30mn) {
        this.setDenomination(denomination);
        this.setPrixLocation30mn(prixLocation30mn);
    }
    public String getDenomination() {return denomination;}
    public void setDenomination(String denomination)
    {this.denomination = denomination;}
    public double getPrixLocation30mn() {return this.prixLocation30mn;}
    public void setPrixLocation30mn(double prixLocation)
    {this.prixLocation30mn = prixLocation;}
}
```

Écrivez pour une sous-classe représentant les cours de gym avec accessoire :

- son entête
- son attribut
- son constructeur avec paramètres permettant d'initialiser tous les attributs
- sa méthode de calcul du prix. Elle doit ajouter, au prix du cours calculé selon sa durée comme expliqué question 2, le prix de location de l'accessoire en tenant compte également de la durée du cours.

```
public class CoursGymAvecAccessoire extends CoursGym { //1
    private Accessoire accessoire; //0,75
    public CoursGymAvecAccessoire(int dureeMn, boolean tapisSolRequis,
        TypeLieu lieu, Accessoire accessoire) { //0,75
        super(dureeMn, tapisSolRequis, lieu); //1
        this.accessoire = accessoire; //0,5
    }
    public double prix() {
        double prix = super.prix() ; //1
        if (accessoire != null) // l'objet est bien initialisé //+0,5 bonus
            prix+= this.accessoire.getPrixLocation30mn()*(this.getDureeMn()/30); //1,5
        return prix ; //0,5
    }
}
```

}