



Nom :	Prénom :
Formation :	Numéro d'étudiant :

Contrôle 1 - Instructions simples et conditionnelles, classes

Tous documents sur support papier autorisés. Durée maximale : 1h20

L'ensemble des réponses sera à donner sur les feuilles d'énoncé. Ne pas dégrafer les feuilles.

Question 1. (2,5 pt) Savoir écrire une séquence d'instructions et une instruction conditionnelle dans un `main`. Remplissez le main du listing 1 afin qu'il donne des indications sur le signe d'un nombre réel, en suivant les 5 étapes ci-dessous :

1. Créer un clavier (objet de la classe `Scanner`)
2. Afficher un message demandant de saisir un nombre réel
3. Saisir au clavier ce nombre réel
4. Si le nombre est positif ou nul, le programme affiche "nombre positif", sinon le programme affiche "nombre négatif"
5. Fermer le clavier

Listing 1 – Programme1.java

```
package controle;  
import java.util.Scanner;  
public class Programme1 {  
    public static void main(String [] args){  
  
        // Solution  
  
        (0,25 pt) Scanner sc = new Scanner(System.in);  
  
        (0,25 pt) System.out.println("veuillez saisir un nombre reel");  
  
        (0,5 pt) double r = sc.nextDouble();  
  
        (0,5 pt)    if (r >= 0)  
        (0,25 pt)        System.out.println("nombre positif");  
        (0,25 pt)    else  
        (0,25 pt)        System.out.println("nombre negatif");  
  
        (0,25 pt)    sc.close();  
  
    }  
}
```

Question 2. (3,5 pt) Savoir utiliser une classe déjà définie. Etudiez le listing de la classe `Materiau` (listing 2). Les matériaux décrits ici sont destinés à la confection de chaussures. Un matériau a un nom, une origine (animale, végétale, synthétique, etc.), une couleur (vert par défaut), une épaisseur en mm et il peut être non velouté.

Listing 2 – `Materiau.java`

```
package controle;
public class Materiau {
    private String nom, origine;
    private Color color = Color.GREEN;
    private double epaisseur; // en mm
    private boolean estVeloute;

    public Materiau() {}
    public Materiau(String nom, String origine, Color color, double epaisseur, boolean estVeloute) {
        this.setNom(nom);    this.setOrigine(origine);
        this.setColor(color); this.setEpaisseur(epaisseur);
        this.setEstVeloute(estVeloute);
    }

    public String getOrigine() {return this.origine;}
    public void setOrigine(String origine) {this.origine = origine;}

    public Color getColor() {return color;}
    public void setColor(Color color) {this.color = color;}

    public double getEpaisseur() {return epaisseur;}
    public void setEpaisseur(double epaisseur) {this.epaisseur = epaisseur;}

    public boolean isEstVeloute() {return estVeloute;}
    public void setEstVeloute(boolean estVeloute) {this.estVeloute = estVeloute;}

    public String getNom() {return nom;}
    public void setNom(String nom) {this.nom = nom;}

    public double epaisseurEnPieds() {return this.epaisseur*0.00328084;}

    public String toString() {
        return "Materiau[_nom=" + nom + ",_origine=" + origine + ",_color=" + color
            + ",_epaisseur=" + epaisseur + ",_estVeloute=" + estVeloute + "];"
    }
}
```

Remplissez le main du listing 3 afin qu'il effectue les opérations suivantes :

1. Créer une instance de la classe `Materiau` représentant du daim velouté, d'origine animale, de couleur `Color.DARK_GRAY`, de 2 mm d'épaisseur.
2. Modifiez l'épaisseur pour qu'elle devienne égale à 3 mm.
3. Afficher l'épaisseur convertie en pieds (au sens de l'unité de mesure) en utilisant la méthode correspondante.

Listing 3 – `Programme2.java`

```
package controle;
public class Programme2 {
    public static void main(String[] args){

        // Solution

        (0,25 pt) Materiau daim =
        (1 pt)      new Materiau("daim", "animale", Color.DARK_GRAY, 2, true);

        (1,25 pt)  daim.setEpaisseur(3);

        (1 pt)    System.out.println("epaisseur_convertie_en_pieds"+daim.epaisseurEnPieds());

    }
}
```

Question 3. Savoir écrire le code d'une nouvelle classe

a- Ecrire l'entête et les attributs d'une classe **Chaussure** représentant le concept de chaussure, avec les informations suivantes. Une chaussure a une référence et un type qui sont des chaînes de caractères, un matériau (de la classe **Matériau** vue à la question précédente; l'attribut vaut **null** si aucun matériau n'est associé), la longueur du pied de la personne (en centimètres), une largeur en notation américaine (AAA, AA, A, B, C, D, E), elle peut ou non avoir des lacets, elle peut avoir ou non des boucles.

Réponse à la question 3.a : 2,5 pt

Listing 4 – Chaussure.java

```
package controle;

// on peut introduire une enumeration pour la largeur americaine ou utiliser le type String
(BONUS de 0,5 pt si c'est fait) public enum TypeLargeur { AAA, AA, A, B, C, D, E }

(0,25 pt) public class Chaussure {

(0,25 pt) private String reference;
(0,25 pt) private String type;
(0,5 pt) private Matériau materiau;
(0,25 pt) private double longueurPied; // en cm
(0,5 pt) private TypeLargeur largeurAmericaine; // ou private String largeurAmericaine;
(0,5 pt) private boolean lacet, boucle;
}
```

b- Ecrire pour la classe **Chaussure** un constructeur qui prenne comme paramètres (1) une référence, (2) un type, (3) un matériau associé, (4) une longueur de pied et qui initialise les attributs correspondants.

Réponse à la question 3.b : 2 pt

Listing 5 – Chaussure.java

```
(1 pt) public Chaussure(String reference, String type, Matériau materiau, double longueurPied) {
(0,25 pt) this.setReference(reference); // ou this.reference = reference;
(0,25 pt) this.setType(type); // ou this.type = type;
(0,25 pt) this.setLongueurPied(longueurPied); // ou this.longueurPied = longueurPied;
(0,25 pt) this.setMatériau(materiau); // ou this.materiau = materiau;
}
```

c- Ecrire pour la classe **Chaussure** l'accessor en modification (**set**) de l'attribut représentant la longueur de pied de la personne. On ne peut accepter que des valeurs comprises entre 5 cm et 42 cm. Si ce n'est pas le cas, on affiche un message d'erreur et l'attribut est inchangé.

Réponse à la question 3.c : 3 pt

Listing 6 – Chaussure.java

```
(1 pt) public void setLongueurPied(double longueurPied) {
    (1 pt) if (longueurPied >= 5 && longueurPied <= 42)
    (0,5 pt)     {this.longueurPied = longueurPied;}
    else
    (0,5 pt)     {System.out.println("une longueur de pied est comprise entre 5 et 42 cm");}
}
```

d- Ecrire pour la classe **Chaussure** une méthode **toString** retournant une chaîne de caractères incluant la référence, le type, et si un matériau lui est associé, le nom et l'épaisseur en pieds (unité de mesure) de ce matériau.

Réponse à la question 3.d 3,5 pt :

Listing 7 – Chaussure.java

```
public String toString() {
    (1 pt) String s = "Chaussure_de_reference=" + reference + ",_type=" + type;
    (1 pt) if (this.materiau != null)
    {
    (0,5 pt)         s += "_fait_en_"+this.materiau.getNom()
                    + "_dont_l'epaisseur_est_de_"+this.materiau.epaisseurEnPieds();
    }
    (1 pt) return s;
}
```

e- Ecrire pour la classe **Chaussure** une méthode **pointure** retournant la pointure correspondant à la longueur du pied de la personne. Pour calculer une approximation de la pointure : prendre la longueur du pied et lui ajouter 1 cm, puis multiplier par $3/2$. Prendre l'arrondi de la valeur résultante. Par exemple, pour une longueur de pied de 25,2 cm, le résultat est la partie entière supérieure de $(25,2 + 1) * 3/2 = 39,3$. C'est-à-dire 40. L'opération `Math.ceil(x)` permet d'obtenir la partie entière supérieure de `x`.

Réponse à la question 3.e : 3 pt

Listing 8 – pointure.java

```
(1 pt) public double pointure() {  
(0,5 pt)     return  
(0,5 pt)         Math.ceil(  
(1 pt)             (this.longueurPied+1)*3/2);  
}
```
