



Nom :  
Prénom :  
Numéro d'étudiant :  
Formation :

## Contrôle 2 - Classes, héritage, ArrayList

Étudiez le listing de la classe Velo (listing 1).

Listing 1 – Velo.java

```
package CC2;
public class Velo {
    private String marque = "";
    private double tailleRoue; //en mm

    public Velo() {}
    public Velo(String m, double tr) {
        this.setMarque(m);
        this.setTailleRoue(tr);
    }

    public String getMarque() {
        return marque;
    }
    public void setMarque(String marque) {
        this.marque = marque;
    }
    public double getTailleRoue() {
        return tailleRoue;
    }
    public void setTailleRoue(double tailleRoue) {
        this.tailleRoue = tailleRoue;
    }
    @Override
    public String toString() {
        return "Velo_{marque=" + marque + ",_tailleRoue=" + tailleRoue + "}";
    }
}
```

### Question 1. Savoir compléter une classe

Réécrire pour la classe Velo la méthode `setTailleRoue(double tailleRoue)` en ajoutant la contrainte suivante : la taille d'une roue (sa circonférence) doit être comprise entre 407 mm et 737 mm. Si ce n'est pas le cas, un message d'erreur est affiché et la valeur donnée est 737.

### Réponse à la question 1 :

// 2 points

```
public void setTailleRoue(double tailleRoue) {
    if (tailleRoue >= 407 && tailleRoue <= 737)
        { this.tailleRoue = tailleRoue; }
    else
        {
            System.out.println("la taille des roues est comprise entre 407 et 737."
                + "la valeur 737 est affectée par défaut");
            this.tailleRoue = 737;
        }
}
```

**Question 2. Savoir compléter une classe**

Ecrire dans la classe `Velo` une méthode calculant le diamètre de la roue d'un vélo. Le diamètre se calcule en divisant la taille (circonférence) par la valeur `Math.PI` ( $\pi$ ).

**Réponse à la question 2 :**

```
// 1 point

public double diametre() {

    return this.getTailleRoue()/Math.PI;

}
```

**Question 3. Savoir écrire le code d'une énumération**

Ecrire une énumération pour les valeurs possibles des types de moteurs de vélo : pédalier, roue, friction.

**Réponse à la question 3 :**

```
// 1 point

public enum TypeMoteurVelo {

    pedaler, roue, friction

}
```

**Question 4. Savoir écrire le code d'une sous-classe**

a- Ecrire l'entête et les attributs d'une classe représentant un *VeloElectrique*. Un vélo électrique, en plus d'une marque et d'une taille de roue, a également un type de moteur (pédalier, roue, ou friction).

**Réponse à la question 4.a :**

```
// 1,5 points

public class VeloElectrique extends Velo {

private TypeMoteurVelo typeMoteur;
.....
}
```

**b-** Ecrire pour la classe *VeloElectrique* un constructeur prenant tous les paramètres nécessaires pour initialiser les attributs d'instance.

**Réponse à la question 4.b :**

```
// 1,5 points

public VeloElectrique(String m, double tr, TypeMoteurVelo typeMoteur) {

    super(m, tr);

    this.typeMoteur = typeMoteur;

}
```

**c-** Redéfinir (en la spécialisant) dans la classe *VeloElectrique* la méthode `toString`. Elle intègre le diamètre de la roue et le type de moteur à la description d'un vélo.

**Réponse à la question 4.c :**

```
// 1,5 points

public String toString() {

    return super.toString()+

        "diametre roue = "+this.diametre()+

        " type moteur = "+this.getTypeMoteur();

}
```

### Question 5. Savoir manipuler une liste d'objets

**a-** Ecrire l'entête et les attributs d'une classe *MagasinVelo* avec les informations suivantes. Un magasin de vélo a un nom et vend des vélos (qui seront placés dans une liste). Initialisez les attributs au moment de leur déclaration. Écrivez un constructeur initialisant le nom du magasin.

**Réponse à la question 5.a :**

```
// 1,5 points

public class MagasinVelo {

    private String nom="";

    private ArrayList<Velo> listeVelosVendus = new ArrayList<>();

    public MagasinVelo(String n) {this.nom=n;}

    ...
}
```

**b-** Ecrire, pour la classe `MagasinVelo`, une méthode `vend` permettant d'ajouter un vélo à la liste des vélos vendus. On doit vérifier que le vélo n'est pas déjà parmi les vélos vendus, sinon un message d'erreur est affiché.

**Réponse à la question 5.b :**

```
// 2 points

public void vend(Velo v) {

    if (listeVelosVendus.contains(v))

        {System.out.println("Ce vélo est déjà dans la liste des vélos vendus");}

    else {this.listeVelosVendus.add(v);}

}
```

**c-** Ecrire, pour la classe `MagasinVelo`, une méthode `nbVelosTailleRoueEntre(int min, int max)` qui retourne le nombre de vélos vendus dont les roues ont une taille strictement supérieure à `min` et inférieure ou égale à `max`.

**Réponse à la question 5.c :**

```
// 2,5 points

public int nbVelosTailleRoueEntre(int min, int max) {

    int nb = 0;

    for (Velo v : this.listeVelosVendus) {

        if (v.getTailleRoue() > min && v.getTailleRoue() <= max) {

            nb ++;

        }

    }

    return nb;

}
```

**d-** Ecrire, pour la classe `MagasinVelo`, une méthode `velosPourEnfant` retournant la liste des vélos vendus dont la taille des roues est inférieure ou égale à 609 mm.

**Réponse à la question 5.d :**

```
// 2,5 points

public ArrayList<Velo> velosPourEnfant(){

    ArrayList<Velo> resultat = new ArrayList<>();

    for (Velo v : this.listeVelosVendus) {

        if (v.getTailleRoue()<= 609) {

            resultat.add(v);

        }

    }

    return resultat;

}
```

**Question 6.** Ecrire un programme `main` dans `MainVelo` où sont créés deux vélos (dont un vélo électrique) en utilisant les constructeurs avec paramètres (inventez les valeurs), un magasin qui vend ces deux vélos et enfin où est affiché le nombre de vélos vendus dont la taille de roue est comprise entre 609 (strictement supérieure à 609) et 737 (inférieure ou égale à 737).

**Réponse à la question 6 :**

```
// 2 points

Velo v1 = new Velo("JoliBike",737);

VeloElectrique v2 = new VeloElectrique("JoliBike",407,TypeMoteurVelo.friction);

MagasinVelo mv = new MagasinVelo("LesVelosDici");

mv.vend(v1);

mv.vend(v2);

System.out.println(mv.nbVelosTailleRoueEntre(609, 737));
```

**Question 7.** Dans le code des classes, nous avons utilisé plusieurs fois des valeurs fixes (407, 609, 737) : 407 représente la plus petite taille de roue, 609 la plus grande taille de roue pour les vélos pour enfants, 737 représente la plus grande taille de roue de vélo. Ces valeurs ne peuvent pas changer.

- Comment représenteriez-vous ces valeurs dans des attributs ? Ecrivez le code Java correspondant en indiquant bien dans quelle classe vous l'écrivez.
- Réécrivez la méthode `setTailleRoue` dans la classe `Velo` en tenant compte de ces attributs.

**Réponse à la question 7 :**

```
// Dans la classe Velo

// 1 point

private final static double plusPetiteTailleRoue = 407;
private final static double plusGrandeTailleRoueEnfant = 609;
private final static double plusGrandeTailleRoue = 737;

// 1 point

public void setTailleRoue2(double tailleRoue) {
    if (tailleRoue >= plusPetiteTailleRoue && tailleRoue <= plusGrandeTailleRoue)
        { this.tailleRoue = tailleRoue; }
    else
        {
            System.out.println("la taille des roues est comprise entre "+
                plusPetiteTailleRoue+" et"+plusGrandeTailleRoue+". "
                + "la valeur "+plusGrandeTailleRoue+" est affectée par défaut");
            this.tailleRoue = plusGrandeTailleRoue;
        }
}
```