

Compte rendu TP5 de POO

Exercice 1

1. Le constructeur de la classe Moteur

```
/**
 *
 * @param cv puissance du moteur
 * @param v v la voiture à laquelle appartient le moteur
 */
public Moteur(int cv, Voiture v){
    puissance=cv;
    voiture=v;
}
```

2. La méthode enPanne() de la classe Moteur

```
/**
 * modifie l'état du moteur à en panne
 */
public void enPanne(){
    enPanne=true;
    demarre=false;
}
```

3. La méthode toString() de la classe Moteur

```
/**
 *
 * @return une chaîne de caractères décrivant le moteur
 */
public String toString(){
    return "[Moteur] "+puissance+" cv,";
}
```

4. Le constructeur de la classe Voiture (attention, celui du CM ne crée pas les 4 roues, mais juste le tableau devant contenir les 4 roues)

```
/**
 *
 * @param cv nombre de chevaux du moteur
 * @param type type de voiture (ex : formule1, twingo, etc)
 * @param tailleRoue taille des roues de la voiture s
 */
public Voiture(int cv, String type, int tailleRoue ){
    modele=type;
    moteur=new Moteur(cv,this);
    roues = new ArrayList();
    for(int i=0;i<4;i++){
        roues.add(new Roue(tailleRoue));
    }
}
```

5. La méthode toString() de la classe Voiture

```
/**
 *
 * @return une chaîne de caractère décrivant la voiture
 */
public String toString(){
    String type = "[Voiture] "+modele;
    String enMarche;
    String res ="";

    if(moteur.estDemarre()==true){ //affiche si le moteur est démarré ou non, et si il est en panne ou non
        enMarche=" est démarré,";
    }else if(moteur.getEnPanne()==true){
        enMarche=" à l'arrêt, en panne.";
    }else{
        enMarche=" à l'arrêt,";
    }

    String mot = "->"+moteur.toString()+enMarche;
    res = type+"\n"+mot;
    if(pilote!=null){ //affiche les détails d conducteur si il y en a un
        String conduct = "->"+pilote.toString();
        res += "\n"+conduct;
    }

    if(remorque!=null){ //affiche les détails de la remorque si il y en a une
        String rem = "->"+remorque.toString();
        res += "\n"+rem;
    }

    String roue="";
    for(int i=0;i<4;i++){
        Roue r =roues.get(i);
        roue += "-> "+r.toString()+" , ";
    }

    res += "\n"+roue;

    return res;
}
```

6. Les méthodes toString() des classes Roue et Remorque

```
/**
 *
 * @return une chaîne de caractères décrivant la roue
 */
@Override
public String toString(){
    return "[Roue] "+taille+" pouces";
}

/**
 *
 * @return une chaîne de caractères décrivant la remorque
 */
public String toString(){
    return "[Remorque] "+capacite+" litres";
}
```

7. La fonction main utilisée et son résultat

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {

    /*AFFICHAGE VOITURE 1*/
    Voiture v1 = new Voiture(600,"Formule1",15);
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 2*/
    Voiture v2 = new Voiture(60,"Twingo",15);
    System.out.println();
    System.out.println(v2.toString());

    /*AFFICHAGE VOITURE 3*/
    Conducteur c1 = new Conducteur("Sylvain");
    v1.changeConducteur(c1);
    v1.demarre();
    System.out.println();
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 4*/
    v1.enPanne();
    System.out.println();
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 5*/
    Remorque r1=new Remorque(300);
    v2.attacheRemorque(r1);
    v2.changeConducteur(c1);
    System.out.println();
    System.out.println(v2.toString());
}
run:
[Voiture] Formule1
->[Moteur] 600 cv, à l'arrêt,
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Twingo
->[Moteur] 60 cv, à l'arrêt,
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Formule1
->[Moteur] 600 cv, est démarré,
->[Conducteur] Sylvain
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Formule1
->[Moteur] 600 cv, à l'arrêt, en panne.
->[Conducteur] Sylvain
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Twingo
->[Moteur] 60 cv, à l'arrêt,
->[Conducteur] Sylvain
->[Remorque] 300 litres
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

```

Exercice 2

8. Écrire une méthode permettant de dupliquer une voiture et tous les éléments dont le cycle de vie est directement associé (Moteur et Roue)

```
/**
 *
 * @param aThis constructeur privé permettant de cloner la voiture
 */
private Voiture(Voiture aThis) {
    this.modele=aThis.modele;
    this.moteur=aThis.moteur;
    this.pilote=aThis.pilote;
    this.remorque=aThis.remorque;
    this.roues=aThis.roues;
}

/**
 *
 * @return un clone de la voiture
 */
public Voiture clone(){
    return new Voiture(this);
}
```

9. Vérifier que vous avez bien fait une duplication à l'aide de test == (devant retourner faux) et equals() (devant retourner vrai si vous l'avez réécrit)

```
run:
[Voiture] Formule1
->[Moteur] 600 cv, à l'arrêt,
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Twingo
->[Moteur] 60 cv, à l'arrêt,
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Formule1
->[Moteur] 600 cv, est démarré,
->[Conducteur] Sylvain
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Formule1
->[Moteur] 600 cv, à l'arrêt, en panne.
->[Conducteur] Sylvain
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Twingo
->[Moteur] 60 cv, à l'arrêt,
->[Conducteur] Sylvain
->[Remorque] 300 litres
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,

[Voiture] Twingo
->[Moteur] 60 cv, à l'arrêt,
->[Conducteur] Sylvain
->[Remorque] 300 litres
-> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces, -> [Roue] 15 pouces,
v2 == vtest : faux
v2 equals vtest : vrai
BUILD SUCCESSFUL (total time: 0 seconds)
```

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {

    /*AFFICHAGE VOITURE 1*/
    Voiture v1 = new Voiture(600,"Formule1",15);
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 2*/
    Voiture v2 = new Voiture(60,"Twingo",15);
    System.out.println();
    System.out.println(v2.toString());

    /*AFFICHAGE VOITURE 3*/
    Conducteur c1 = new Conducteur("Sylvain");
    v1.changeConducteur(c1);
    v1.demarre();
    System.out.println();
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 4*/
    v1.enPanne();
    System.out.println();
    System.out.println(v1.toString());

    /*AFFICHAGE VOITURE 5*/
    Remorque r1=new Remorque(300);
    v2.attacheRemorque(r1);
    v2.changeConducteur(c1);
    System.out.println();
    System.out.println(v2.toString());

    /*AFFICHAGE VOITURE 6 : CLONE DE VOIURE 5*/
    Voiture vtest = v2.clone();
    System.out.println();
    System.out.println(vtest.toString());

    if(v2==vtest){
        System.out.println("v2 == vtest : vrai");
    }else{
        System.out.println("v2 == vtest : faux");
    }

    if(v2.equals(vtest)){
        System.out.println("v2 equals vtest : vrai");
    }else{
        System.out.println("v2 equals vtest : faux");
    }
}

```