

Solutions

Solution 1.

- D'après la documentation de l'API Java, les méthodes `chainesEgales`, `occurrence` et `extraire` s'appellent respectivement (dans la classe `String`) `equals`, `indexOf` et `substring`.
- La version récursive des méthodes `extraire` et `nombreOccurences` est la suivante :

```
public class Chaine {

    public static String extraire(String s, int debut, int fin) {
        if(debut == fin)
            return "";
        return s.charAt(debut) + extraire(s, debut + 1, fin);
    }

    public static int nombreOccurences(char l, String mot) {
        if (mot.length() == 0)
            return 0;
        return (mot.charAt(0) == l ? 1 + nombreOccurences(l, mot.substring(1)) :
            nombreOccurences(l, mot.substring(1)));
    }

    public static void main(String[] args) {
        String test = "toto";
        String test1 = "totos";
        System.out.println("extraire de '1 3' dans 'toto' : " +
            extraire(test, 1, 3));
        System.out.println("t dans 'toto' : " + nombreOccurences('t', test));
    }
}
```

Solution 2.

1. 1ère version de `MobileSurAxe` :

```
public class MobileSurAxe {

    private static final int POSITION_INITIALE = 0;
    private static final int INCREMENT = 1;
    private static final int DECREMENT = 1;

    private int position;

    public MobileSurAxe(int position) {
        this.position = position;
    }
}
```

```

public MobileSurAxe() {
    this(POSITION_INITIALE);
}

public void avancer() {
    position += INCREMENT;
}

public void reculer() {
    position -= DECREMENT;
}

public int position() {
    return position;
}
}

```

2. 2ème version de MobileSurAxe :

```

public class MobileSurAxe {

    private static final int POSITION_INITIALE = 0;
    private static final int INCREMENT = 1;
    private static final int DECREMENT = 1;

    private int position;
    private int decrement;
    private int increment;

    public MobileSurAxe(int position, int increment,
                       int decrement) {
        this.position = position;
        this.increment = increment;
        this.decrement = decrement;
    }

    public MobileSurAxe(int position) {
        this(POSITION_INITIALE, INCREMENT, DECREMENT);
    }

    public MobileSurAxe() {
        this(POSITION_INITIALE);
    }

    public void avancer() {
        position += INCREMENT;
    }
}

```

```

public void reculer() {
    position -= DECREMENT;
}

public int position() {
    return position;
}

public void fixerIncrement(int nouveauIncrement) {
    increment = nouveauIncrement;
}

public void fixerDecrement(int nouveauDecrement) {
    increment = nouveauDecrement;
}

public int lireIncrement() {
    return increment;
}

public int lireDecrement() {
    return decrement;
}
}

```

3. on ajoute la méthode suivante :

```

public void afficherEtat() {
    System.out.println("Position : " + position + " increment : "
        + increment + " decrement : " + decrement);
}

```

On peut aussi surcharger la méthode `toString` de la façon suivante :

```

public String toString() {
    return ("Position : " + position + " increment : "
        + increment + " decrement : " + decrement);
}

```

L'affichage est alors effectué de la façon suivante :

```

MobileSurAxe m1 = new MobileSurAxe();
System.out.print(m1);

```

4. Programme de test :

```

public class Test {

```

```

public static void main(String[] args) {

    MobileSurAxe m1 = new MobileSurAxe(1,3,1);
    MobileSurAxe m2 = new MobileSurAxe(31,5,5);

    while(m1.position() < m2.position()) {
        m1.avancer();
        System.out.print("m1_-->");
        m1.afficherEtat ();

        m2.reculer ();
        System.out.print("m2_-->");
        m2.afficherEtat ();
    }
}

```

5. Il faut ajouter un attribut *numero* et un attribut statique *compteur* pour que les numéros soient distincts.

```

public class MobileSurAxe {

    private static final int POSITION_INITIALE = 0;
    private static final int INCREMENT = 1;
    private static final int DECREMENT = 1;

    private int position;
    private int decrement;
    private int increment;
    private int numero;
    private static int compteur = 0;

    public MobileSurAxe(int position, int increment,
                       int decrement) {
        this.position = position;
        this.increment = increment;
        this.decrement = decrement;
        numero = compteur++;
    }

    public MobileSurAxe(int position) {
        this(POSITION_INITIALE, INCREMENT, DECREMENT);
    }

    public MobileSurAxe() {
        this(POSITION_INITIALE);
    }
}

```

```

    }

    public void avancer() {
        position += increment;
    }

    public void reculer() {
        position -= decrement;
    }

    public int position() {
        return position;
    }

    public void fixerIncrement(int nouveauIncrement) {
        increment = nouveauIncrement;
    }

    public void fixerDecrement(int nouveauDecrement) {
        decrement = nouveauDecrement;
    }

    public int lireIncrement() {
        return increment;
    }

    public int lireDecrement() {
        return decrement;
    }

    public void afficherEtat() {
        System.out.println("Position : " + position + " _increment : "
            + increment + " _decrement : " + decrement);
    }

    public int numero() {
        return numero;
    }
}

```