

Corrigés des premiers exercices sur les structures récursives

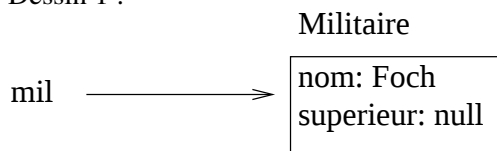
Exercice 8.1.1 Dessins et chemins

On reprend en l'enrichissant le premier exemple de classe récursive du cours. Il s'agit d'une structure contenant des militaires. Chaque militaire est identifié par son nom et il a un supérieur qui est lui aussi un militaire.

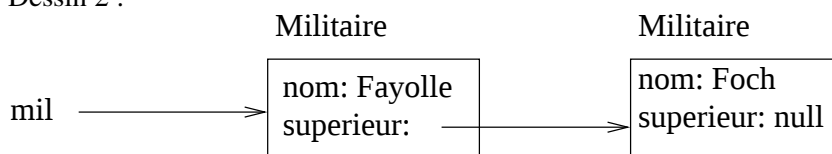
Question 1 dessins

Pour chacun des points marqués par un commentaire dans la méthode `main`, dessinez la structure construite et la variable `mil`. Vous ferez des dessins analogues à ceux du cours. Pour simplifier, vous pouvez ne pas représenter les objets de type `String`. Attention : la dernière instruction de `main` est un appel de constructeur sans affectation. Son effet est différent des deux instructions précédentes.

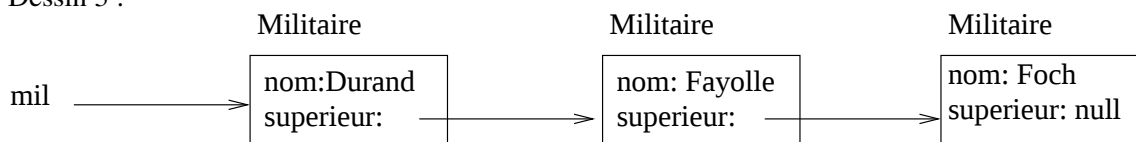
Dessin 1 :



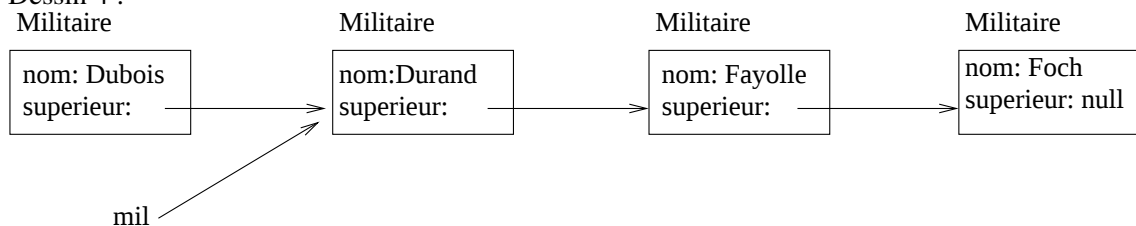
Dessin 2 :



Dessin 3 :



Dessin 4 :
Militaire



Question 2 *chemins*

1. quelles instructions faut-il ajouter à la fin de la méthode `main` pour afficher à l'écran le nom du général Fayolle (en allant chercher ce nom dans l'objet créé à la troisième ligne de la méthode `main`).

```
Terminal.ecrireStringln(mil.getSuperieur().getNom());
```

2. quelles instructions faut-il ajouter à la fin de la méthode `main` pour affecter à une variable `chef` l'objet représentant le maréchal Foch ?

```
Militaire chef=mil.getSuperieur().getSuperieur();
```

3. quelles instructions faut-il ajouter à la fin de la méthode `main` pour afficher le nom du soldat Dubois ?

Réponse : il est impossible d'accéder à l'objet qui contient le nom Dubois. Aucune instruction Java ne peut utiliser cet objet.

Exercice 8.1.2 *affichage de militaires*

Question 1 *méthode d'affichage*

Ajoutez à la classe `Militaire` de l'exercice précédent une méthode d'affichage qui affiche toute la structure.

Question 2 *méthode récursive*

Même question mais en écrivant une méthode récursive (sans boucle).

Question 3 *virgules*

Reprendre la méthode de la première question et séparer les militaires de la structure au moyen de virgules. Attention, il ne doit pas y avoir de virgule avant le premier ni après le dernier militaire affiché.

```
public void afficher1(){ // question 1
    Militaire tmp = this;
    while(tmp != null){
        Terminal.ecrireString(tmp.getNom() + " ");
        tmp = tmp.getSuperieur();
    }
    Terminal.sautDeLigne();
}
public void afficher2(){ // question 2
    Terminal.ecrireString(nom + " ");
    if (superieur != null)
        superieur.afficher2();
    else
        Terminal.sautDeLigne();
}
public void afficher3(){ // question 3
```

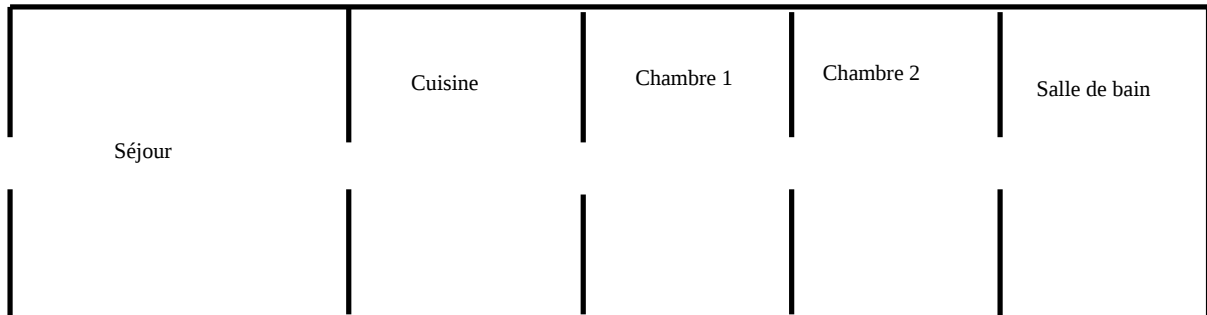
```

    Militaire tmp = this;
    while(tmp != null){
        Terminal.ecrireString(tmp.getNom());
        if (tmp.getSuperieur() != null)
            Terminal.ecrireString(", ");
        tmp = tmp.getSuperieur();
    }
    Terminal.sautDeLigne();
}
public static void main(String[] args){
    Militaire mil=new Militaire("Foch");
    // dessin 1
    mil = new Militaire("Fayolle",mil);
    // dessin 2
    mil = new Militaire("Durand",mil);
    // dessin 3
    new Militaire("Dubois",mil);
    // dessin 4
    mil.afficher1();
    mil.afficher2();
    mil.afficher3();
}

```

Exercice 8.1.3 Pièces d'une maison

Les longères (maisons traditionnelles) sont trop étroites pour comporter un couloir. Toutes les pièces sont en enfilades. Dans cet exercice, on suppose que l'entrée est sur un des pignons (petits côtés) de la maison. Voici un plan donnant un exemple de longère.



On veut représenter chaque pièce avec son nom d'usage (cuisine, chambre, salon), un numéro d'ordre (pour distinguer les chambres par exemple, chambre 1 et chambre 2) et une référence vers la pièce suivante.

Ecrivez une classe pour représenter les pièces d'une longère et dans la méthode main, construisez une structure décrivant le plan donné ci-dessus.

```

public class Piece{
    private String type;
    private int rang=0;
    private Piece suivante;
    public Piece(String n, Piece s){
        type = n;

```

```
        suivante = s;
    }
    public Piece(String n, int r, Piece s){
        type = n;
        rang = r;
        suivante = s;
    }
    public String getType(){
        return type;
    }
    public int getRang(){
        return rang;
    }
    public Piece getSuivante(){
        return suivante;
    }
    public String getIdent(){
        return type + "_" + rang;
    }
    public void affiche(){
        if (rang != 0)
            Terminal.ecrireString(getIdent()+"_");
        else
            Terminal.ecrireString(type + "_");
        if (suivante == null)
            Terminal.sautDeLigne();
        else
            suivante.affiche();
    }
    public static void main(String[] args){
        Piece longere;
        longere = new Piece("Salle_de_bain",null);
        longere = new Piece("Chambre",2,longere);
        longere = new Piece("Chambre",1,longere);
        longere = new Piece("Cuisine",longere);
        longere = new Piece("Séjour",longere);
        longere.affiche();
    }
}
```
