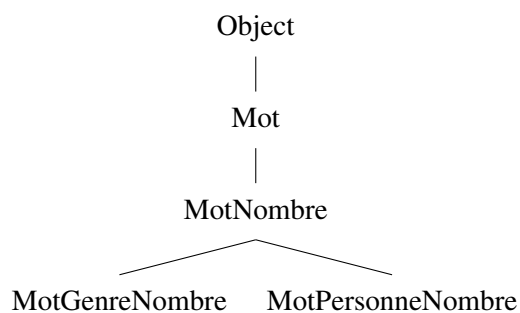


# Énoncés 7

## Exercices sur le typage et la surcharge

### 1 Exercice de typage

On reprend les classes du cours : `Mot`, `MotNombre`, `MotGenreNombre` et `MotPersonneNombre` avec l'arborescence suivante.



Voici un morceau de code avec des inconnues (XXX, YYY, ZZZ et TTT) à remplacer.

---

```
public static void main(String[] args){
    XXX mot1 = new YYY;
    ZZZ mot2 = new TTT;
    System.out.println(mot1.getForme());
    System.out.println(mot2.getForme());
    System.out.println(mot2.estSingulier());
}
```

---

Donnez toutes les combinaisons de classes possibles pour XXX, YYY, ZZZ et TTT (vous ne vous préoccupez pas des paramètres des constructeurs pour YYY et TTT).

### 2 Exercice sur les constructeurs

#### Question 1

Voici un programme qui illustre le comportement des constructeurs avec des appels au constructeur de la super-classe (`super`). Ce programme se focalise sur les constructeurs, il n'y a rien d'autre dans les classes.

---

```
class Premiere{
```

```

    Premiere(){
        Terminal.ecrireStringln("constructeur_de_Premiere");
    }
}
class Seconde extends Premiere{
    Seconde(boolean b){
        super();
        Terminal.ecrireStringln("constructeur_de_Seconde");
    }
}
class Troisieme extends Premiere{
    Troisieme(int i){
        super();
        Terminal.ecrireStringln("constructeur_de_Troisieme");
    }
}
class Quatrieme extends Troisieme{
    Quatrieme(double d){
        super(14);
        Terminal.ecrireStringln("constructeur_de_Quatrieme");
    }
}
class Exo19_1_1{
    public static void main(String[] args){
        new Premiere();
        Terminal.ecrireStringln("=====");
        new Seconde(true);
        Terminal.ecrireStringln("=====");
        new Troisieme(15);
        Terminal.ecrireStringln("=====");
        new Quatrieme(12.3);
    }
}

```

1. dessinez le graphe d'héritage de ce programme, retraçant les relations d'héritage entre classes sans oublier `Object`, la super-classe de `Premiere`.
2. prédisez ce que ce programme affiche.
3. compilez et exécutez ce programme pour vérifier vos prédictions.
4. y a-t-il un constructeur qui s'exécute sans afficher de message à la création de certains objets ?

## Question 2

```

class Cinquieme extends Premiere{
    Cinquieme(){
        Terminal.ecrireStringln("constructeur_de_Cinquieme");
    }
}
class Sixieme extends Cinquieme{
    Sixieme(){
        Terminal.ecrireStringln("constructeur_de_Sixieme");
    }
}

```

```

    }
}
class Exo19_1_2{
    public static void main(String[] args){
        new Cinquieme();
        Terminal.ecrireStringln("=====");
        new Sixieme();
    }
}

```

---

1. prédisez ce que ce programme affiche.
2. compilez et exécutez ce programme pour vérifier vos prédictions.

### Question 3

Le programme suivant provoque une erreur à la compilation : la classe Huitieme est incorrecte.

```

class Septieme extends Premiere{
    Septieme(int i){
        Terminal.ecrireStringln("constructeur_de_Septieme");
    }
}
class Huitieme extends Septieme{
    Huitieme(){
        Terminal.ecrireStringln("constructeur_de_Huitieme");
    }
}

```

---

Essayez de trouver l'erreur. Si vous n'y arrivez pas, essayez de compiler ce programme : le message d'erreur peut vous mettre sur la voie.

### Question 4

```

class Neuvieme extends Premiere{
    Neuvieme(int i){
        Terminal.ecrireStringln("premier_constructeur_de_Neuvieme");
    }
    Neuvieme(boolean b){
        Terminal.ecrireStringln("second_constructeur_de_Neuvieme");
    }
}
class Dixieme extends Neuvieme{
    Dixieme(double d){
        super(true);
        Terminal.ecrireStringln("premier_constructeur_de_Dixieme");
    }
    Dixieme(int x, boolean y){
        super(x);
        Terminal.ecrireStringln("second_constructeur_de_Dixieme");
    }
}
class Exo19_1_4{

```

```
public static void main(String[] args){
    new Dixieme(10.5);
    Terminal.ecrireStringln("=====");
    new Dixieme(45, true);
}
}
```

---

1. prédisez ce que ce programme affiche.
2. compilez et exécutez ce programme pour vérifier vos prédictions.

### 3 Exercice d'héritage : volaille

Un éleveur de volaille reçoit d'un fournisseur de jeunes canards et de jeunes poulets qu'il élève jusqu'à ce qu'ils aient la taille nécessaire à leur commercialisation.

Une volaille est caractérisée par son poids et un numéro d'identification reporté sur une bague qu'elle porte à sa petite patte. Les volailles arrivent à l'élevage à l'âge de trois semaines. Elles sont baguées et enregistrées dans le système informatique.

Il y a deux sortes de volailles : des canards et des poulets. Le prix du canard et celui du poulet sont deux prix différents, exprimés en euros par kilo. En revanche, le prix est le même pour tous les individus de la même espèce. Ce prix varie chaque jour. Le poids auquel on abat les bêtes est différents pour les canards et les poulets, mais c'est le même pour tous les poulets (respectivement, tous les canards).

Ecrivez une classe des volailles avec deux sous-classes des poulets et des canards. Il faut pouvoir enregistrer les prix du jour, les poids d'abattage, le poids d'une volaille donnée.

Écrivez une classe permettant de représenter l'ensemble des animaux de l'élevage au moyen d'un tableau. Des méthodes doivent permettre de trier les animaux à abattre et d'évaluer le prix obtenu pour ces animaux. Il faut également pouvoir enregistrer les jeunes animaux qui arrivent.