

Travaux Dirigés #1:

Exercice 1 :

Prenez comme exemple une de vos activités sportives, culturelles, artistiques ou sociales, et faites une liste des objets impliqués dans cette activité. Dans un premier temps, créez ces objets sous forme de couples attribut/valeur. Dans un deuxième temps, réfléchissez au lien d'interaction existant entre ces objets ainsi qu'à la manière dont ils sont capables de s'influencer mutuellement. Dans un troisième temps, identifiez pour chaque objet une possible classe le caractérisant.

Exercice 2 :

Répondez aux questions suivantes :

- Un même référent peut-il désigner plusieurs objets ?
- Plusieurs référents peuvent-ils désigner un même objet ?
- Un objet peut-il faire référence à un autre ? Si oui, comment ?
- Pourquoi l'objet a-t-il besoin d'une classe pour exister ?
- Un objet peut-il changer d'état ? Si oui, comment ?
- Que signifie cette écriture $a.f(x)$?
- Où doit être déclarée $f(x)$ pour que l'instruction précédente s'exécute sans problème ?
- Qu'appelle-t-on un envoi de message ?
- Comment un premier objet peut-il conduire un deuxième objet à changer d'état ?

Exercice 3 :

Réalisez, en Java, un programme contenant une classe `Point`, avec ses trois coordonnées dans l'espace (x,y,z) , et que l'on peut initialiser de trois manières différentes (on connaît soit x , soit x et y , soit x et y et z). Ensuite, intégrez dans la classe une méthode `translate()` qui est surchargée trois fois, dépendant également de celles des trois valeurs des translations qui sont connues.

Exercice 4 :

Créez deux objets de la classe `Point` de l'exercice précédent (Exercice 3) et testez le bon fonctionnement du programme quand vous translatez ces points.

Exercice 5 :

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(){
        System.out.println("constructeur de Premiere");
    }
}
class Seconde extends Premiere{
    Seconde(boolean b){
        super();
        System.out.println("constructeur de Seconde");
    }
}
class Troisieme extends Premiere{
    Troisieme(int i){
        super();
        System.out.println("constructeur de Troisieme");
    }
}
class Quatrieme extends Troisieme{
    Quatrieme(double d){
```

```

        super(14);
        System.out.println("constructeur de Quatrieme");
    }
}

class Exo1{
    public static void main(String[] args){
        new Premiere();
        System.out.println("=====");
        new Seconde(true);
        System.out.println("=====");
        new Troisieme(15);
        System.out.println("=====");
        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 Première.
2. Qu'affiche ce programme ?
3. y a-t-il un constructeur qui s'exécute sans afficher de message à la création de certains objets ?

Exercice 6 :

```

class Cinquieme extends Premiere{
    Cinquieme(){
        System.out.println("constructeur de Cinquieme");
    }
}
class Sixieme extends Cinquieme{
    Sixieme(){
        System.out.println("constructeur de Sixieme");
    }
}
class Exo2{
    public static void main(String[] args){
        new Cinquieme();
        System.out.println("=====");
        new Sixieme();
    }
}

```

1. Qu'affiche ce programme ?

Exercice 7 :

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

```

class Septieme extends Premiere{
    Septieme(int i){
        System.out.println("constructeur de Septieme");
    }
}
class Huitieme extends Septieme{
    Huitieme(){
        System.out.println("constructeur de Huitieme");
    }
}

```

1. Qu'affiche ce programme ? Essayez de trouver l'erreur.

Exercice 8 :

```
class Neuvieme extends Premiere{
    Neuvieme(int i){
        System.out.println("premier constructeur de Neuvieme");
    }
    Neuvieme(boolean b){
        System.out.println("second constructeur de Neuvieme");
    }
}
class Dixieme extends Neuvieme{
    Dixieme(double d){
        super(true);
        System.out.println("premier constructeur de Dixieme");
    }
    Dixieme(int x, boolean y){
        super(x);
        System.out.println("second constructeur de Dixieme");
    }
}
class Exo4{
    public static void main(String[] args){
        new Dixieme(10.5);
        Terminal.ecrireStringln("=====");
        new Dixieme(45, true);
    }
}
```

1. Qu'affiche ce programme ?