

Licence Informatique 2ème année

Programmation Objet 1

TD n°1 – Modularité

1- Mon premier programme objet

1.1 Écrire un programme objet qui affiche "Hello world". Ce programme devra créer un objet et l'affichage se fait par appel d'une méthode de cet objet.

1.2 Même question, mais avec deux objets o_1 et o_2 . La méthode main appelle une méthode $m()$ de o_1 (méthode sans paramètre), qui appelle une méthode de o_2 , qui affiche "Hello world".

2- Objets et méthodes

Qu'affichent les programmes A et B?

```
class A {  
    int a = 0;  
    void f() { a++; }  
    void affiche() { System.out.println(a); }  
    public static void main(String[] args) {  
        A p = new A();  
        A q = new A();  
        p.affiche(); q.affiche();  
        p.f();  
        p.affiche(); q.affiche();  
        p = q;  
        p.f();  
        p.affiche(); q.affiche();  
    }  
}
```

```
class B {  
    int a = 0;  
    void f(B o) {  
        o.a++;  
        o.affiche();  
    }  
    void affiche() { System.out.println(a); }  
    public static void main(String[] args) {  
        B obj = new B();  
        obj.affiche();  
        obj.f(obj);  
        obj.affiche();  
    }  
}
```

3- Soyons rationnels

Écrire une classe permettant de manipuler des nombres rationnels. La classe permettra :

- de créer un rationnel avec la valeur 0 par défaut
- de créer un rationnel dont la valeur est un entier passé en paramètre
- de créer un rationnel à partir du numérateur et du dénominateur
- de créer un rationnel à partir d'un autre rationnel
- de modifier la valeur du numérateur et du dénominateur
- d'afficher le rationnel sous forme de fraction ou sous forme décimale
- d'additionner deux rationnels
- de multiplier deux rationnels, ou un rationnel et un entier
- écrire un main qui crée deux rationnels 3/4 et 7 et affiche leur somme

4- Égalité d'objets

Donner la sortie du programme Java suivant :

```
class Truc {
    int i;
    Truc(int a) {
        i = a;
    }
    Truc(Truc t) {
        i = t.i;
    }
    public boolean equals(Object t) {
        if (t instanceof Truc) return (((Truc) t).i == i);
        else return false;
    }
    public static void main(String args[]) {
        Truc y = new Truc(1);
        Truc z = y;
        Truc w = new Truc(y);
        if(z == y) System.out.println(1);
        if(w == y) System.out.println(2);
        if(z.equals(y)) System.out.println(3);
        if(w.equals(y)) System.out.println(4);
    }
}
```

5- Création et destruction d'objet

```
class Truc {
    Truc() { System.out.println("Cet objet est tout le portrait de sa classe"); }
    protected void finalize() { System.out.println("Requiem in pace, brave Truc"); }
}
```

Donner la sortie des programmes suivants :

```

5.1  public static void main(String args[]) {
        Truc x;
    }

5.2  public static void main(String args[]) {
        Truc x = new Truc();
    }

5.3  public static void main(String args[]) {
        Truc x = new Truc();
        System.gc();
    }

5.4  public static void main(String args[]) {
        Truc x = new Truc();
        x = new Truc();
        System.gc();
    }

```

6- Tagada tagada, voilà les Dalton

On veut écrire un programme objet pour gérer les personnages des Dalton dans un jeu vidéo. Les Dalton ont chacun un nom et une taille : Joe (137cm), Jack (165cm), William (190cm) et Averell (210cm). On veut faire parler les Dalton de façon uniforme en appelant une méthode `String parle()` qui renvoie ce que dit le Dalton, mais chaque Dalton s'exprime différemment : Joe répète *Je hais Lucky Luke*, Jack et William disent toujours *Calme toi Joe* et Averell dit toujours *Quand est-ce qu'on mange?*.

6.1 Écrire un programme objet qui permet de créer les objets correspondant aux Dalton.

6.2 Donner un exemple de code qui crée les Dalton et les fait parler.

Chaque Dalton doit être repéré par une position (en mètres) dans un univers à deux dimensions, une direction de déplacement (vecteur en mètres) et une vitesse (en mètre par seconde). Le déplacement des Dalton doit être géré de manière uniforme et doit permettre de modifier la direction de déplacement de chaque Dalton, et de faire se déplacer un Dalton pendant un certain temps. Par exemple, si Joe a une vitesse de 1.5m/s, le faire se déplacer pendant 60s modifiera sa position de 90m dans la direction de déplacement¹.

En outre chaque Dalton peut utiliser, ou non, un cheval. Dans le jeu, un cheval a un nom, une couleur de pelage (on peut utiliser la classe `Color` de Java), une position, une direction et une vitesse de déplacement (en m/s). On peut déplacer un cheval de la même façon qu'on déplace un Dalton, mais si un Dalton est sur un cheval, les deux changent de direction et de position ensemble. Dans ce cas, la vitesse de déplacement est évidemment celle du cheval.

6.3 Écrire du code objet et/ou modifier le code précédent pour ajouter les chevaux dans le programme et permettre de lier ou de délier un Dalton et un cheval et de gérer les déplacements des Daltons et des chevaux.

¹ Rappel : un Dalton se trouvant à la position (x, y) avec une direction (dx, dy) et une vitesse v se retrouvera, au bout d'un temps t , à la position $\left(x + \frac{dx \times v \times t}{\sqrt{dx^2 + dy^2}}, y + \frac{dy \times v \times t}{\sqrt{dx^2 + dy^2}}\right)$