FICHE BILAN
INITIATION A LA PROGRAMMATION ORIENTEE OBJET

1. DESCRIPTION DE LA PROGRAMMATION ORIENTEE OBJET

OBJETS

D'après wikipaedia la programmation orientée objet (POO) est un paradigme de programmation informatique. Il consiste en la définition et l'interaction de briques logicielles appelées *objets*.

Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations.

L'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes.

CLASSES

En programmation orientée objet, la déclaration d'une classe regroupe des membres comportant des méthodes et des attributs (propriétés) communs à un ensemble d'objets.

La classe déclare, d'une part, des attributs représentant l'état des objets et, d'autre part, des méthodes représentant leur comportement.

Une classe représente donc une catégorie d'objets. Elle apparaît aussi comme un moule ou une usine à partir de laquelle il est possible de créer des objets. Il s'agit en quelque sorte d'une « boîte à outils » qui permet de fabriquer un objet.

L'action de créer un objet à partir d'une classe est appelé : « instanciation ». On parle alors d'un objet en tant qu'instance d'une classe (création d'un objet ayant les propriétés de la classe).

STRUCTURE D'UNE CLASSE

De manière générale, chaque objet est construit sur un modèle appelé « classe ». Ce modèle doit comporter :

- Les paramètres caractérisant l'objet qui sont nommés « **attributs** »,
- Les actions que peut réaliser l'objet, qui sont nommés « **méthodes** »,
- Un « **constructeur »** qui permet de créer l'espace mémoire nécessaire à la vie de l'objet, ainsi que de donner des valeurs particulières à chaque attribut de l'objet, le rendant ainsi unique. Par définition, le constructeur est une méthode qui porte le même nom que la classe.

Toutes les **classes** auront la même architecture : **Attributs / Constructeur / Méthodes**.

Lorsqu'on crée un objet en instanciant une classe, on fait appel au constructeur de la classe en précisant des valeurs particulières à chaque attribut de la classe. Une fois l'objet instancié, il est possible d'accéder à ses attributs et à ses méthodes pour le faire vivre.

2. UN EXEMPLE : CLASSE « BALLE »

MODELE :

Classe « balle »

PARAMETRES

Attributs de la balle :

- Couleur : c
- Diamètre : d
- Position :
  - Abscisse : x
  - Ordonnée : y
- Déplacement :
  - Déplacement suivant x : dx
  - Déplacement suivant y : dy

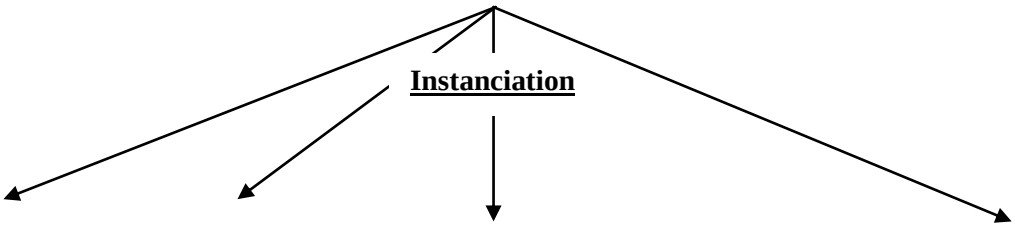
CONSTRUCTEUR

balle()

ACTIONS

Méthodes de la balle :

- afficher()
- avancer()
- testCollision()



balle1

- c = rouge
- d = 20 pix
- x = 100 pix
- y =150 pix
- dx = +3 pix
- dy = -1pix

Balle2

- c = vert
- d = 35 pix
- x = 110 pix
- y =250 pix
- dx = 1 pix
- dy = 2 pix

Balle3

- c = bleu
- d = 20 pix
- x = 200 pix
- y =200 pix
- dx = 2 pix
- dy = 1 pix

…

balleN

- c = …
- d = … pix
- x = … pix
- y = … pix
- dx = … pix
- dy = … pix

PROGRAMMATION :

```
class balle {
        // Attributs
        int x, y, dx, dy, d ;
        color c ;

        // Constructeur
        balle(…){…}

        // Méthodes
        void afficher{…}
        void avancer{…}
        void rebondir{…}
}
```

3. LE CONSTRUCTEUR

ROLE DU CONSTRUCTEUR

Le rôle du constructeur est :

- de créer l'espace mémoire nécessaire à la vie de l'objet, c'est-à-dire un espace suffisant pour contenir à la fois les attributs et les méthodes.
- d'initialiser chaque attribut de l'objet, avec des valeurs particulières, le rendant ainsi unique.

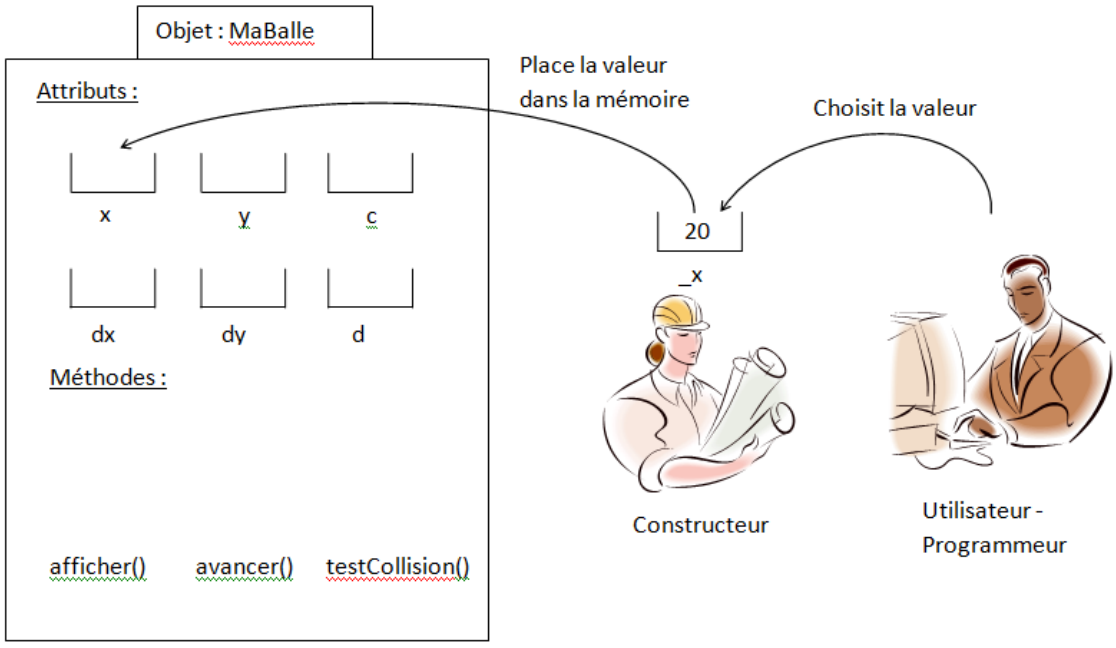
Par définition, le constructeur est une méthode qui porte le même nom que la classe. La différence avec une méthode de la classe est que le constructeur ne revoit aucun type : son écriture n'est précédée par rien.

Pour lui permettre d'initialiser les attributs de la classe, le constructeur doit comporter autant de paramètres d'entrée (appelés « arguments ») que d'attributs. Les arguments du constructeur ne peuvent pas porter le même nom que l'attribut correspondant. Dans le cas contraire, le compilateur de Processing sera incapable de différencier les deux variables.

EXEMPLE DE LA CLASSE « BALLE »

Les noms des arguments sont résumés dans le tableau suivant :

Type	Attributs de la classe	Arguments du constructeur
int	x	_x
int	y	_y
int	dx	_dx
int	dy	_dy
int	d	_d
color	c	_c



PROGRAMMATION DU CONSTRUCTEUR :

```
balle(int _x, int _y, int _dx, int _dy, int _d, color _c){
    x = _x;
    y = _y;
   dx = _dx;
   dy = _dy;
   d = _d;
   c = _c;
}
```

4. PROGRAMMATION D'UN OBJET

DECLARATION DE L'OBJET AVANT LE SETUP()

```
balle maBalle; // déclaration de l' objet de type « balle »
```

INSTANCIATION DE L'OBJET DANS LE SETUP()

```
void setup(){
…
 maBalle = new balle( 100, //x = 100 pixels
                      150, //y = 150 pixels
                      1, //dx = 1 pixels entre deux frames
                      2, //dy = 2 pixels entre deux frames
                      20,// d = 20 pix de diamètre
                      color(255,0,0)) ; // couleur rouge
…
}
```

Ici l'instanciation de l'objet « maBalle » se fait grâce à l'opérateur ***new*** suivi du nom de la classe à instancier et des parenthèses contenant les paramètres du constructeur.

UTILISATION DE L'OBJET DANS LE DRAW() ET AUTRES METHODES

```
void draw(){
…
   // affichage de la balle
    maBalle.afficher() ;
  // faire avancer la balle
    maBalle.avancer() ;
     // Test de collision sur les bords
    maBalle.testCollision() ;
…
}
```